(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 665 682 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
12.09.2001 Bulletin 2001/37

(51) Int Cl.[7]: **H04N 1/60**

(21) Application number: **95300572.5**

(22) Date of filing: **30.01.1995**

(54) **Colour image processing method**

Farbbildverarbeitungsverfahren

Procédé de traitement d'images en couleurs

(84) Designated Contracting States:
DE ES FR GB IT NL

(30) Priority: **31.01.1994 JP 1002394**

(43) Date of publication of application:
02.08.1995 Bulletin 1995/31

(60) Divisional application:
99203472.8 / 0 981 244

(73) Proprietor: **CANON KABUSHIKI KAISHA**
Tokyo (JP)

(72) Inventor: **Kumada, Shuichi, c/o Canon K.K.**
Ohta-ku, Tokyo (JP)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
High Holborn
2-5 Warwick Court
London WC1R 5DJ (GB)

(56) References cited:
EP-A- 0 448 250
EP-A- 0 546 773
EP-A- 0 555 866
EP-A- 0 574 905
GB-A- 2 201 562
US-A- 5 058 040
US-A- 5 185 661

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

[0001] The present invention relates to a color gamut displaying method, an image processing method, and apparatus capable of utilizing the methods, which are suitable for a color management system, and the like, for instance, reproducing colors faithfully in consideration with characteristics of a color input/output device.

[0002] As shown in Fig. 32, color ranges which can be reproduced by devices, such as a scanner, a monitor, and a printer, are different from each other.

[0003] In Fig. 32, x and y denotes axes for chromaticity, and insides of closed areas are color gamuts.

[0004] In other words, the key to reproduce colors after considering the difference in a color gamut is how to deal with portions of ranges which are not common to different devices. For example, when transformation from a monitor color space to a printer color space is considered, a major problem is that which color should be used to express a color outside of a color gamut of a printer. A process to deal with the above-said problem is called a color gamut compression process.

[0005] A color management system has major functions, such as a function of performing the color gamut compression process for faithfully reproducing colors taking into consideration the characteristics of a color input/output device, as well as a color gamut checking function for checking whether or not it is possible to reproduce a color using an output device. The latter function is used to determine whether or not color data, included in an application software and the like, can be reproduced by an output device (e.g., printer monitor).

[0006] Specifically, a method which is shown in Fig. 34 is suggested by the present inventor.

[0007] According to the method, a color gamut of a printer is defined by a color reproduction solid whose shape is a hexahedron, determined by peaks of red, green, blue, cyan, magenta, yellow, white, and black. Then each surface of the hexahedron is divided into two triangles to obtain total of twelve areas. Finally, it is determined whether image data is inside or outside of the areas, thereby determining if the image data is in the color gamut or not.

[0008] However, there are problems to be solved in the aforesaid method.

(1) A real color gamut of a printer has a three dimensional solid which is too complicated to express with a hexahedron, thus preciseness of checking the color gamut is low.

(2) It is not a simple process to determine whether or not image data is inside of the twelve areas, thus it requires a long time to complete the process.

[0009] US-A-5185661 describes an input scanner and print engine system which corrects for color interpretation errors in the input scanner's color filter set and which maps the input colors into the color gamut of a printer.

[0010] EP-A-0546773 describes a graphical user interface for displaying the boundary of a color gamut and for interactively editing a color palette.

[0011] EP-A-0574905 describes a method and apparatus for forming color images wherein a first color correction is performed to obtain a target color chromatically equal to the input color and printable by a printer and a second color correction is performed when it is determined that the input color is not reproducible by the printer so as to correct the input color into the visually nearest target color within the color gamut of the printer.

[0012] According to one aspect of the present invention, there is provided a color image processing method comprising the steps of:

storing a color gamut table representing the colors which can be reproduced by a color reproduction device arranged to produce a color reproduction of color image data input to the device; and
determining whether a color can be reproduced by the color reproduction device by using the stored color gamut table, characterised by carrying out the storing step by storing the color gamut table so that each address within the table is associated with a respective different color of the color input image data and the data stored at each address has either a first value indicating that the associated color falls within the color gamut or a second value indicating that the associated color falls outside the color gamut and carrying out the determining step by using the color image data to be input to the color reproduction device to address the color gamut table to determine whether the first or second value is stored at the address corresponding to that color and so to determine whether or not the color represented by the color data falls within the color gamut of the color reproduction device.

[0013] In another aspect, the present invention provides a color image processing apparatus comprising:

storage means storing a color gamut table representing the colors which can be reproduced by a color reproduction device arranged to produce a color reproduction of color image data input to the device; and
determination means for determining whether a color can be reproduced by the color reproduction device by using the stored color gamut table, characterised in that each address within the color gamut table is associated with a respective different color of the color input image data and the data stored at each address has either a first value indicating that the associated color falls within the color gamut or a second value indicating that the associ-

ated color falls outside the color gamut and in that the determination means is arranged to determine whether a color can be reproduced by the color reproduction device by using the color image data to be input to the color reproduction device to address the color gamut table to determine whether the first or second value is stored at the address corresponding to that color and so to determine whether or not the color represented by the color data falls within the color gamut of the color reproduction device.

[0014] Apparatus and a method embodying the invention enable a color gamut expression for realizing a color gamut checking function which improves preciseness of checking and shortens a processing time in a color management system and the like.

[0015] An image processing method and an apparatus utilizing the method embodying the invention are capable of converting input image data into image data which enables input image data to be reproduced more faithfully by a color reproduction device.

[0016] An image processing method and apparatus embodying the invention are most suitable for a checking function which checks input image data and a color gamut of a printer, and reproduce colors designated by the input image data more faithfully.

[0017] According to a constitution as described above, there can be provided an image processing method capable of converting input image data into image data which enables input image data to be reproduced more faithfully by a color reproduction device.

[0018] Further, a color gamut for realizing the color gamut checking function is easily confirmed, and the preciseness of determining the color gamut can be improved, as well as the time to process the color reproduction can be shortened.

[0019] Embodiments of the present invention will now be described, by way of example, with reference to the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

Fig. 1 is a block diagram showing a configuration of an image processing system;
Fig. 2 is a flowchart showing a process to output an image by a color printer after applying color reproduction process to input color image data;
Fig. 3 is an equation in matrix notation, showing relationship between XYZ data, defined by CIE, and RGB data;
Fig. 4 is an equation in matrix notation, showing relationship between the RGB data, defined by CIE, and XYZ data;
Fig. 5 explains an example of converting the XYZ data, defined by the CIE, into L* a* b*;
Fig. 6 is an example of a three dimensional LUT (look up table) used for color gamut compression;
Fig. 7 is an expanded view of one of a cubic including input values of L*, a*, and b* to the LUT in Fig. 6;
Fig. 8 is a flowchart showing a process to display an image on a color monitor after applying color reproduction process to input color image data;
Fig. 9 is a flowchart showing a process in which whether or not the color image data to be processed is within a color gamut of an output device is determined, and the image is either displayed or not displayed on a color monitor on the basis of the determined result;
Fig. 10 is an example of a color gamut of a printer inside of a L* a* b* solid;
Fig. 11 is an example showing a case where a color gamut of a monitor or a printer is defined by an area in a two dimensional plane of a* and b*, constructed with rectangular lattices, which is obtained by slicing a three dimensional color solid in the direction perpendicular to the L* axis according to a first example not falling within the scope of the invention claimed;
Fig. 12 shows an example of a in a case where the gamut of a monitor or a printer is defined by an area constructed with rectangular lattices according to the first example;
Fig. 13 is a flowchart illustrating a process of checking the color gamut on the basis of the color gamut data table in Fig. 12;
Fig. 14 is a flowchart illustrating a process of checking the color gamut on the basis of the color gamut data table in Fig. 12;
Fig. 15 shows an example in a case where the color gamut of a monitor or a printer is defined by the two dimensional range of a* and b*, constructed with rectangular lattices, which is obtained by slicing the three dimensional color solid in the direction perpendicular to the L* axis according to a second example not falling within the scope of the invention claimed;
Fig. 16 is an address table where a top address of the color gamut data table in Fig. 17 is stored according to the second example;
Fig. 17 is an example of a color gamut data table in a case where the gamut of a monitor or a printer is defined in two dimensional coordinate range according to the second example;
Fig. 18 is a color gamut data table in a case where $\Delta a = \Delta b = 1$ in Fig. 15 according to the second example;
Fig. 19 is a flowchart showing a process of checking the color gamut on the basis of the tables, shown in Figs. 16 and 17, which are for displaying the color gamuts according to the second example;
Fig. 20 is a flowchart showing a process of checking the color gamut on the basis of the tables, shown in Figs. 16 and 17, which are for displaying the color gamuts according to the second example;
Fig. 21 is a flowchart showing a process of checking

the color gamut on the basis of the tables, shown in Figs. 16 and 17, which are for displaying the color gamuts according to the second example;

Fig. 22 is an example of a case where a gamut of a monitor or a printer is defined by bit map in the two dimensional area of a* and b*, constructed with rectangular lattices, which is obtained by slicing the three dimensional color solid in the direction perpendicular to the L* axis according to an embodiment of the present invention;

Fig. 23 is an example of a bit map table in a case where the gamut in Fig. 22 is defined by bit map according to the embodiment;

Fig. 24 shows an address table storing a top address of the color gamut data table in Fig. 23, as an example, according to the embodiment;

Fig. 25 is a flowchart showing a process of checking the color gamut on the basis of the tables, shown in Figs. 23 and 24, for displaying the color gamuts according to the embodiment;

Fig. 26 is a flowchart showing a process of checking the color gamut on the basis of the tables, shown in Figs. 23 and 24, for displaying the color gamuts according to the embodiment;

Fig. 27 is an example of a case where a gamut of a monitor or a printer is defined by an area, surrounded by coordinate points and lines which connect the coordinate points in the two dimensional area of a* and b*, constructed with rectangular lattices, which is obtained by slicing the three dimensional color solid in the direction perpendicular to the L* axis according to a third example not falling within the scope of the invention claimed;

Fig. 28 is a color gamut data table in a case where a reproduction area of a monitor or a printer is defined by an area which is surrounded by coordinate points and lines which connect the coordinate points according to the third example;

Fig. 29 is a flowchart of checking the color gamut on the basis of the color gamut data table shown in Fig. 28 according to the third example;

Fig. 30 is a flowchart of checking the color gamut on the basis of the color gamut data table shown in Fig. 28 according to the third example;

Fig. 31 is an example of color reproduction by a color management system;

Fig. 32 is an example showing differences in color gamut among devices; and

Fig. 33 shows an example of a color gamut compression;

Fig. 34 shows an example of a conventional method of checking color reproduction of a device.

[0020] Fig. 31 shows an example of color reproductionusing a color management system. As a color input/output device in Fig. 31, a basic color scanner, color monitor, and color printer are considered.

[0021] In the color management system, a color reproduction process is performed in consideration with characteristics of those three devices. The process is composed of following four color space conversion processes, shown by arrows in Fig. 31.

[0022] 1. Conversion from a color space of the scanner to a color space of the monitor

[0023] A process to display data of a photograph read by the scanner onto the monitor faithfully.

[0024] 2. Conversion from a color space of the scanner to a color space of the printer

[0025] A process to print data of a photograph read by the scanner by the printer faithfully.

[0026] 3. Conversion from a color space of the monitor to a color space of the printer

[0027] A process to print data, such as character and figure data, generated on the monitor by using an application software or the like, by a printer faithfully.

[0028] 4. Conversion from a color space of the printer to a color space of the monitor

[0029] A process to preview an image to be printed by a printer on the monitor.

[0030] Note that the aforesaid color spaces rely on the three devices, and the color gamuts differ from each other.

[0031] Following is an example of a conversion process from the color space of the monitor to the color space of the printer, as described above in 3.

[0032] Fig. 1 is a block diagram showing a circuit configuration of an image processing system.

[0033] As shown in Fig. 1, the image processing system comprises a color monitor 1, host computer 2, and color printer 3. The host computer 2 process image data, instructs the color monitor 1 to display the processed result and also instructs the color printer 3 to print.

[0034] The host computer 2 comprises a monitor interface 21 for interchanging data with the color monitor 1; VRAM 22 for maintaining display data used for monitor display; CPU 23 including ROM, RAM, and the like, for controlling the entire system; a frame memory 24 used for temporary storing image information to be displayed on the monitor; a print buffer 25 used for printer output; a printer interface 26 for interchanging data with the color printer 3; a color gamut information memory 27 for storing information for displaying a color gamut, such as a color gamut data table; a color gamut checking unit 28 for determining whether or not input color image data is in the color gamut; color converter 29 for converting color spaces relating color reproduction; and color reproduction processor 2a for reproducing color on the basis of the determined result by the color gamut checking unit 28.

[0035] A process to apply a color reproduction to input color image data and to output the processed image to the color printer 3 by the host computer 2 having aforesaid constitution, more specifically by the color reproduction processor 2a, is described below with reference to a flowchart in Fig. 2.

[0036] In Fig. 2, the host computer 2 reads a set of

input color image data to be processed (luminance data of R, G, B) which is generated at step S10 in advance or sent from another device, such as the scanner, then the process moves to step S11. Note that the input color image data to be processed in the present example depends on an input device, such as a scanner, and the data is RGB data defined based on colorimetry (values of chromaticity of R, G, B, and white are definite).

[0037] The read RGB data is converted into data in CIE 1931 standard colorimetric system (indicated as "XYZ data", hereinafter) which is defined by CIE (Commission Internationale de l'Eclairage) by a color converter 29 at step S11, the process proceeds to step S12. At step S12, the XYZ data, converted by the color converter 29 at step S11, is further converted into L*a*b* data, then the process goes to step S13. At step S13, the L*a*b* data is converted into CMY (cyan, magenta, yellow) data for a printer, and the process moves to step S14.

[0038] At step S14, the CMY data is stored in the print buffer 25, then the process proceeds to step S15. At step S15, whether or not there is any remaining input color image data is checked. If there is, the process goes back to step S10.

[0039] Whereas if there is not, the process moves to step S16, and the CMY data, stored in the print buffer 25, is sent to the color printer 3 via the printer I/F 26. After an image is printed out by the color printer 3, the process is completed.

[0040] As described above, the input color image data to be processed is RGB data defined based on colorimetry (values of chromaticity of R, G, B, and white are definite), thus equations shown in Figs. 3 and 4 show the relationship between RGB data and XYZ data which is defined by the CIE. Therefore, the equations in Figs. 3 and 4 can be used in a process at step S11 in Fig. 2. P_{ij}(i, j = 1, 2, 3) in Fig. 3 and Q_{ij}(i, j, = 1, 2, 3) in Fig. 4 are constants defined based on colorimetry of input color image data.

[0041] Fig. 5 is an example of a process at step S12 in Fig. 2. Xn, Yn, and Zn in Fig. 5 are determined depending upon a kind of used CIE standard light source.

[0042] Step S13 in Fig. 2 is a conversion process, including color gamut compression, to convert the L*a*b* data including colors beyond the color gamut of a printer to the CMY data which is within the color gamut of the printer.

[0043] There are several methods of color gamut compression.

[0044] Fig. 33 shows an example of color gamut compression.

[0045] Methods of color gamut compression are to convert image data to color data which can be expressed by an output device. Three general examples are described below.

[0046] First method is to use a sense of human beings as shown in left lower part in Fig. 33. This is to set the lightest and the darkest colors of the image data to the lightest and the darkest colors which can be expressed by the output device, respectively, and to convert other colors of the image data so that they are expressed in relative darkness between the lightest and the darkest colors to be expressed by the output device.

[0047] Second method is to maintain the colors in the common color gamut of the image data and the output device, and to convert the color of image data outside of the common color gamut onto the border of the gamut of the output device with the intensity being maintained.

[0048] Third method is to maintain chromaticity of the colors outside of the common color gamut of the image data and the output device, and to compress them.

[0049] Further, there is a method which uses a three dimensional LUT (look up table) as one of the color gamut compression methods, and the LUT which is used for conversion by color gamut compression at step S13 according to the embodiment is shown in Fig. 6. The example of the LUT is a three dimensional table composed of a plurality of rectangular solids formed by dividing a color solid in a three dimensional space of L*a*b* in the each coordinate direction at a uniform interval (ΔL^*, Δa^*, Δb^*). At each intersection of lattices, namely each corner of the rectangular solid, a value of CMY corresponding to a sampling value of L*a*b* is stored.

[0050] Fig. 7 is an expanded view of one of the rectangular solids including a L*a*b* value inputted to the LUT in Fig. 6. Note that the z direction is L* axis, x direction is a* axis, and y direction is b* axis. Assume that the input L*a*b* value is at a point P.

[0051] When a left lower front point of the lattice is (x_i, y_i, z_i) and $\Delta L^* = \Delta a^* = \Delta b^* = 1$, and the displacement from the left lower front point to the point P is x_f, y_f, z_f, then the coordinate points are shown in Fig. 7. Here, if the CMY value, stored at the each point of intersection of lattices, is denoted as U(x, y, z), then the CYM value of the point P, $U(x_i + x_f, y_i + y_f, z_i + z_f)$ can be obtained by utilizing a adding method as shown by following equations.

[Equation 1]

[0052]

$$
\begin{aligned}
& U(x_i + x_f, y_i + y_f, z_i + z_f) \\
&= U(x_i, y_i, z_i) \mathrm{x} (1 - x_f)(1\ Y_f)(1 - z_f) \\
&+ U(xi + 1, y_i, z_i) \times x_f (1 - yf)(1 - z_f) \\
&+ U(x_i, y_i + 1, z_i) \times (1 - x_f)\, yf\, (1 - z_f) \\
&+ U(x_i, y_i, z_i + 1) \times (1 - x_f)(1 - yf)\, z_f \\
&+ U(x_i, y_i + 1, z_i + 1) \times (1 - x_f)\, yf\, z_f \\
&+ U(x_i + 1, y_i, z_i + 1) \times x_f (1 - yf)\, z_f
\end{aligned}
$$

$$+ U(x_i + 1, yi + 1, z_i) \times x_f\, y_f\, (1 - z_f)$$

$$+ U(x_i + 1, yi + 1, z_i + 1) \times x_f\, y_f\, z_f$$

[0053] As above, the CMY value corresponding to the input L*a*b* value can be obtained by using the LUT and the adding process.

[0054] By performing the aforesaid color gamut compression, an image can be reproduced by a printer more faithfully to the original.

[0055] Next, a process to transmit data to the color monitor 1 after applying the color reproduction to the input color image data at the host computer 2, especially, the color reproduction processor 2a, will be described below with reference to a flowchart in Fig. 8.

[0056] The host computer 2 reads a set of input color image data (luminance data for RGB) at step S20, as at step S10 in Fig. 2, then the process proceeds to step S21. There, the input color image data which is RGB data defined based on colorimetry (values of chromaticity of R, G, B, and white are definite) depends on the kind of input device.

[0057] At step S21, the read RGB data is converted into XYZ data, determined by CIE, in the color converter 29, then the process proceeds to step S22. At step S22, the color converter 29 further converts the XYZ data into RGB data depending on characteristics of a color monitor, after that the process moves to step S23. At step S23, the RGB data for a monitor is stored in the frame memory 24, and the process proceeds to step S24. At step S24, whether or not there is remaining input color data is checked. If there is, the process goes back to step S20.

[0058] Whereas, if there is no input color image data left, the process moves to step S25, and the RGB data for monitor which is stored in the frame memory is displayed on the color monitor 1 by using the VRAM 22, and the process is completed.

[0059] Since the input color image data and the monitor RGB data are RGB data defined based on colorimetry (values of chromaticity of R, G, B, and white are definite), there is relationship between aforesaid data and XYZ data determined by CIE as shown by equations in Figs. 3 and 4. It is possible to apply the relationship equation in Fig. 3 in the process at step S21, and the relationship equation in Fig. 4, in the process at step S22. Pij (i, j = 1, 2, 3) and Qij (i, j, = 1, 2, 3) are constants determined on the basis of colorimetry of input color image data and of monitor RGB for the color monitor 1.

[0060] Next, referring to Fig. 9, there will be described a characteristic process in which it is checked on the color monitor 1 whether or not color image data (for example, R, G, B luminance data which is generated by utilizing an application software or the like) to be processed is in a color gamut of the color monitor 1 or the color printer 3, and an image is displayed on the color monitor 1 in accordance with the checked result.

[0061] Fig. 9 is a flowchart showing a process of checking the color gamut, in the host computer 2, especially in the color gamut check unit 28, in which it is checked whether or not data, obtained by transforming color image data (R, G, B luminance data) which is generated by using an application software or the like on the color monitor 1 so as to be in a color space of an output device, such as the color monitor 1 or of the color printer 3, is in a color gamut of the output device, and a process to display an image on the color monitor 1 in accordance with the checked result.

[0062] In Fig. 9, the host computer 2 reads color (a set of R, G, B luminance data) to be checked for a color gamut at step S30, and the process moves to step S31. Note that RGB data which depends on the characteristics of the color monitor 1, and defined on the basis of colorimetry (values of chromaticity of R, G, B, and white are definite) is used as the color to be checked.

[0063] At step S31, the read RGB data at step S30 is converted into XYZ data, defined by CIE, by the color converter 29, then the process moves to step S32. At step S32, the XYZ data is further converted into L*a*b* data by the color converter 29, and the process proceeds to step S33. At step S33, whether or not the converted L*a*b* data can be reproduced by the color printer 3 is checked. If so, a flag is turned ON, whereas, if not, a flag is turned OFF, and the process moves to step S34.

[0064] At step S34, whether the flag set at step S33 is ON or OFF is checked. If the flag is not ON, the process moves to step S36, and the checked color is displayed in white, and a process is completed. Whereas, if the flag is ON, since it is possible to reproduce colors without processing any further, the process moves to step S35, and the checked color is displayed on the color monitor 1 without being processed, then the process is completed.

[0065] Aforesaid process is applied to all sets of the input image data.

[0066] Therefore, in a case where the color to be checked is within the color gamut of an output device, such as the color monitor 1 or of the color monitor 3, the checked color is faithfully reproduced on the color monitor 1. Whereas in a case where the checked color is outside of the color gamut, the checked color is displayed in white on the color monitor 1 to inform a user, so that the user can recognize which color at which part is outside of the color gamut.

[0067] Note that the checked color is represented by white when the color is outside of a color gamut. However, it is not limited to white as long as the user knows that the checked color is outside of the color gamut, thus black can be used instead of white, for instance.

[0068] As seen in Fig. 10, the aforesaid range of lattice which includes the L*a*b* value inputted to the LUT shown in Fig. 6 differs depending on the value of L*. Therefore, feature of the embodiment is to have data of the range of lattice as color reproduction information and

is how to use the information to perform color gamut check. The feature will be described below. It should be noted that values of ΔL^*, Δa^*, and Δb^* can be the minimum value which L^*, dispersion data, can take as a predetermined interval.

(First Example not falling within the scope of the invention claimed)

[0069] According to a first example, a gamut of a printer, one of output devices, is defined by an area including all the area inside of a bold line in Fig. 11, and the color gamut is checked based on the defined area information. Let the area be defined by two points, (a_{max}, b_{max}) and (a_{min}, b_{min}). Since the size of the area changes depending on the value of L^*, an example of a color gamut data table in the first example is shown in Fig. 12. In this example, tables are stored in the color gamut information memory 27.

[0070] An area surrounded by bold line in Fig. 11 shows the color gamut.

[0071] Note that a number L within L^*_{min} and L^*_{max} is m (constant).

[0072] Figs. 13 and 14 are flowcharts of checking the color gamut on the basis of the color gamut data table in Fig. 12 by the host computer 2, especially by the color reproduction checking unit 28.

[0073] First, the host computer 2 sets a pointer at the top of the color gamut data table in Fig. 12 at step S100, and the process proceeds to step S101. At step S101, a constant n is set to 1 and L1 is set to L^*_{min}, then the process moves to step S102. At step S102, the read L^* is compared with L_1. If the values of L^* and L1 are not identical, then the process moves to step S103, thereat L_{n+1} is set to $L_n + \Delta L$, then the process further moves to step S104. At step S104, values of n and pointer are incremented by 1, and process goes back to step S102.

[0074] Whereas, if the read L^* has the same value as L_1, then the process proceeds to step S105, and the read a^* is compared with a_{nmin}. If a^* is smaller than a_{nmin}, then the process moves to step S106, thereat a flag is set OFF and the process is completed.

[0075] RGB data of color to be checked is RGB data defined on the basis of colorimetry (values of chromaticity of R, G, B, and white are definite), therefore, the equations in Figs. 3 and 4 are held to represent relationship between the RGB data and the XYZ data which is defined by CIE. It is possible to use the equation in Fig. 3 in the process at step S31. P_{ij} (i, j = 1, 2, 3) and Q_{ij} (i, j, = 1, 2, 3) are constants depending on definition based on colorimetry of monitor RGB for the color monitor 1.

[0076] A process at step S32 in Fig. 9 is the same as the aforesaid process in Fig. 5. Note that Xn, Yn, and Zn are the value depending on type of CIE standard light source to be used.

[0077] At step S33 in Fig. 9, $L^*a^*b^*$ data of the color to be checked which is obtained at step S32, is checked whether or not it is within color gamut, and the result is outputted as a value of a flag. A color gamut of a printer is included in a $L^*a^*b^*$ color solid, and as shown in Fig. 10, the range can be designated by using the three dimensional solids which are produced by dividing the color solid of $L^*a^*b^*$ three dimensions in the each coordinate direction at a fixed interval (ΔL^*, Δa^*, Δb^*), whose process is the same as the one in Fig. 6.

[0078] Further, regarding a color gamut of a monitor, although the range differs from the range of a printer, the range can be designated by using the three dimensional lattices which are produced when the color solid of $L^*a^*b^*$ three dimensions is divided in the each coordinate direction at a fixed interval (ΔL^*, Δa^*, Δb^*), similarly.

[0079] When the three dimensional color solid is sliced in the direction which is perpendicular to L^*, the cut surface is the two dimensional plane of a^* and b^* which is constructed with rectangular (square, in the first example) areas. Lengths of sides of each rectangular area are Δa^* and Δb^*, as shown in Fig. 11. Note that the area surrounded by a closed bold line indicates a range of lattices including the color gamut of a monitor or a printer in the sliced two dimensional plane perpendicular to L^*.

[0080] Therefore, in accordance with whether or not the three dimensional lattices including input image data are within the designated range, whether or not the input image data is within a color gamut of a color reproduction device is determined.

[0081] Whereas, if it is determined at step S105 that a^* is not smaller than a_{nmin}, then the process moves to step S107, thereat read a^* is compared with a_{nmax}. If a^* is larger than a_{nmax}, then the process moves to step S106, where a flag is set OFF and the process is completed.

[0082] If a^* is not larger than a_{nmax} at step S107, then the process proceeds to step S108, where read b^* and b_{nmin} are compared. If b^* is smaller than b_{nmin}, the process moves to step S106, thereat a flag is set OFF and the process is completed.

[0083] Whereas if it is determined at step S108 that b^* is not smaller than bnmin, then the process proceeds to step S109, and the read b^* and b_{nmax} are compared. If b^* is larger than b_{nmax}, the process moves to step S106, thereat a flag is set OFF and the process is completed.

[0084] If b^* is not larger than b_{nmax}, then the process proceeds to step S110, and the process is completed after a flag is set ON.

[0085] Accordingly, a color gamut is easily checked based on the color gamut data table in Fig, 12.

(Second Example)

[0086] According to a second example, a color gamut of a printer or a monitor is defined by a two dimensional coordinate range including all the area inside of a bold line in Fig. 15, and the color gamut is checked based on

the defined area information. In Fig. 15, the coordinate ranges is decided in accordance with amin and a_{max} in each b* lattice range in lattice range between b^*_{min} and b^*_{max}. These tables indicating the color gamuts are stored in the color gamut information memory 27.

[0087] Fig. 16 is an address table which stores top addresses of the aforesaid color gamut data table shown in Fig. 17 in the second example. As shown in Fig. 16, in the address table of the second example, a number L within a range between L^*_{min} and L^*_{max} is m (constant).

[0088] In the color gamut data table in Fig. 17, information on the range of a* in each lattice within a range between b^*_{min} and b^*_{max} is stored. Note that, as shown in Fig. 15, for example, there can be two a* ranges corresponding to a range between b^*_{min} and $b^*_{min} + \Delta b$, and likewise there can be a case where there are a plurality of a* ranges in a arbitrary b* lattice range. Constants in Fig. 17, $i_1, i_2, ..., i_n$ correspond to a number in the a* range in each b* lattice range.

[0089] Fig. 18 is a color gamut data table representing the color gamut of a printer shown in Fig. 15, when $\Delta a = \Delta b = 1$.

[0090] Figs. 19 to 21 are flowcharts showing a process for checking a color gamut on the basis of a information table for representing a color gamut described in Figs. 16 and 17 by the host computer 2, especially the color gamut checking unit 28. A process of checking a color gamut in the second example will be described below with reference to Figs. 19 to 21.

[0091] First, at step S200 in Fig. 19, a pointer is set at the top of the address table shown in Fig. 16, and the process moves to step S201. At step S201, a constant n is set to 1 and L_1 is set to L^*_{min}, then the process further proceeds to step S202. Successively at step S202, the read L* is compared with L_1. If the values of L* and L_1 are not identical, then the process moves to step S203, thereat L_{n+1} is set to $L_n + \Delta L$ and the process proceeds to step S204. At step S204, n and pointer are incremented by 1, and the process goes back to step S202.

[0092] Whereas, if L*, read at step S202, and L_1 have the same value, the process proceeds to step S205, thereat an address which is stored in the area that the pointer indicates is fetched, then the process further proceeds to step S206. At step S206, the pointer is set to the fetched address, and the process moves to step S207 in Fig. 20. At step S207, the read b* is compared with b_{min}. If b* is smaller than b_{min}, then the process moves to step S208 in Fig. 21, thereat a flag is set OFF to complete the process.

[0093] Whereas, if b* is not smaller than b_{min} at step S207, then the process moves to step S209, thereat the read b* is compared with b_{max}. If b* is larger than b_{max}, then the process moves to step S208 in Fig. 21, thereat a flag is set OFF to complete the process.

[0094] If it is determined at step S209 that b* is not larger than b_{max}, then the process proceeds to step S210, thereat the constant m is set to 1 and b1 is set to b^*_{min}, and the process moves to S211. At step S211, b_{m+1} is set to $b_m + \Delta b$, and the process proceeds to step S212. At step S212, whether or not the read b* fulfills a relationship, $bm \le b^* < b_{m+1}$, is checked. If not, the process proceeds to step S213, thereat m and pointer are incremented by 1, then the process goes back to step S211.

[0095] If the read b* fulfills the relationship, $b_m \le b^* < b_{m+1}$, then the process moves to step S214 in Fig. 21, thereat the constant j is set to 1, and the process proceeds to step S215. At step S215, the read a* is compared with a_{mjmin}. If a* is smaller than the a_{mjmin}, the process moves to step S208, thereat a flag is set OFF to complete the process.

[0096] Whereas, if it is determined at step S215 that a* is not smaller than a_{mjmin}, then the process proceeds to step S216, thereat the read a* and a_{mjmin} are compared. If a* is larger than a_{mjmin}, the process moves to step S208, thereat a flag is set OFF to complete the process.

[0097] If it is determined at step S216 that a* is not larger than a_{mjmin}, the process proceeds to step S217, thereat i_n is compared with j. If i_n and j have the same value, then the process moves to step S219, where a flag is set ON, and the process is completed.

[0098] Whereas, if it is determined at step S217 that i_n and j do not have the same value, then the process proceeds to step S218, thereat j is incremented by 1, and the process goes back to step S215.

[0099] As described above, a color reproduction can be easily checked by using information tables in Figs. 16 and 17 representing a color gamuts.

(Embodiment)

[0100] According to an embodiment of the present invention, a gamut of a monitor or a printer is defined by bit map tables corresponding to a lattice area, and the color gamut is checked based on the defined bit map information. In the third embodiment, a bit map table is a color gamut data table, and its top is the bit corresponding to a lattice including the left uppermost point, (a_{min}, b_{max}), in Fig. 22. In Fig. 22, let an area surrounded by a bold line be a color gamut of a monitor or a printer, and let the bits outside of the color gamut be OFF, and the bits inside of the range be ON, the bit map table corresponding a case shown in Fig. 22 is represented as in Fig. 23.

[0101] Note that the size of a coordinate range differs depending on a value of L*, therefore color gamut data tables for respective L*'s differ from each other. Fig. 24 is an address table where the top address of a color gamut data table, such as the one shown in Fig. 23, is stored. In Fig. 24, the number of L within the range between L^*_{min} and L^*_{max} is m (constant).

[0102] The tables shown in Figs. 23 and 24 are stored in the color gamut information memory 27.

[0103] Figs. 25 and 26 are flowcharts for checking the

color gamut on the basis of information tables representing the color gamut data tables in Figs. 23 and 24 by the host computer 2, especially by the color reproduction check unit 28. Below, the process to check the color gamut according to this embodiment will be described with reference to Figs. 25 and 26.

[0104] In this embodiment, first, at step S300 in Fig. 25, a pointer is set at the top of an address table in Fig. 24, then the process proceeds to step S301. At step S301, the constant n is set to 1 and L_1 is set to L^*_{min}, and the process moves to step S302. At step S302, the read L* is compared with L_1. If the values of L* and L_1 is not the same, the process moves to step S303, thereat L_{n+1} is set to $L_n + \Delta L$, and the process further moves to step S304. At step S304, n and the pointer are incremented by 1, then the process goes back to step S302.

[0105] Whereas, if it is determined at step S302 that L* and L_1 have the same value, then the process proceeds to step S305, thereat an address which is stored in the area that the pointer indicates is fetched, and the process further proceeds to step S306. A pointer is set to the address fetched at step S306, then the process moves to step S307 in Fig. 26. At step S307, by using a* and b*, a value obtained by dividing $(a^* - a^*_{min})$ by Δa is set to a constant p, and a value obtained by dividing $(b^*_{max} - b^*)$ by Δb is set to a constant q, then the process moves to step S308.

[0106] At step S308, a value obtained by adding 1 to the integer part of p is set to a constant r, a value obtained by adding 1 to the integer part of q is set to a constant s, then the process proceeds to step S309. At step S309, the pointer is moved by r bits to the right and by s bits to down, and the process moves to step S310. Then at step S310, the bit indicated by the pointer is checked. In a case where the bit indicated by the pointer is not ON, the process moves to step S311, thereat a flag is set OFF to complete the process.

[0107] If the bit indicated by the pointer is ON, the process moves to step S312, thereat the flag is set ON to complete the process.

[0108] As described above, the color gamut can be easily checked on the basis of the information tables in Figs. 23 and 24 representing the color gamuts.

(Third Example not falling within the scope of the invention claimed)

[0109] In a third example, there is described a process in which a gamut of a monitor or a printer is defined by an area surrounded by coordinate points as shown in Fig. 27, then a color gamut is checked on the basis of the information on the area.

[0110] Fig. 27 shows an example of a gamut in L^*_{min} which is expressed with the 20 coordinate points in this example. Since the size of the surrounded area changes depending on a value of L*, an example of a color gamut data table which corresponds to the value of L* is shown in Fig. 28 in this example. The range of L* is from L^*_{min} to L^*_{max}, and there are values, in the table, of coordinate points corresponding to the various L*'s. A number of L in the range between L^*_{min} and L^*_{max} is m (constant). Further, a number of constants $i_1, i_2, ..., i_n$ is as same as the number of coordinate points each of which is corresponding each value of L*. The table in this example is stored in the color gamut information memory 27.

[0111] Figs. 29 and 30 are flowcharts of checking the color gamut on the basis of the color gamut data table in Fig. 28 by the host computer 2, especially by the color reproduction check unit 28. The process of checking the color gamut in this example will be explained with reference to flowcharts in Figs. 29 and 30 below.

[0112] In this example, first, a pointer is set at the top of the color gamut data table in Fig. 28 at step S400 in Fig. 29, and the process proceeds to step S401. At step S401, a constant j is set to 1 and L^*_{min} is set to L_1, then the process moves to step S402. At step S402, the read value of L* is compared with L_1. If the values of L* and L_1 are not equal, the process moves to step S403, thereat $Lj + \Delta L$ is set to L_{j+1}, and the process proceeds to step S404. At step S404, j and the pointer is incremented by 1, and the process goes back to step S402.

[0113] Whereas, if L* and L_1 have the same value, the process proceeds to step S405, thereat a number of a coordinate point r is fetched from an area where the pointer designates, then the process moves to step S406. At step S406, a constant k is set to 1, then the process moves to step S407 in Fig. 30. At step S407, $a_{j\cdot k}$ and $b_{j\cdot k}$ are fetched from an area where the pointer designates and the process proceeds to step S408. Then k is compared with r at step S408. If r is larger than k, then the process proceeds to step S409, and after k is incremented by 1, the process goes back to step S407.

[0114] If it is determined at step s408 that r is not larger than k, the process proceeds to step S410, and whether or not a* and b* are included in a closed area surrounded by coordinate points, $(a_{j\cdot 1}, b_{j\cdot 1})$, $(a_{j\cdot 2}, b_{j\cdot 2})$, ..., $(a_{j\cdot k}, b_{j\cdot k})$, fetched at step S411 is checked. If not, the process proceeds to step S412, thereat a flag is set OFF, and the process is completed.

[0115] Whereas, if a* and b* are included in the closed area surrounded by coordinate points, $(a_{j\cdot 1}, b_{j\cdot 1})$, $(a_{j\cdot 2}, b_{j\cdot 2})$, ..., $(a_{j\cdot k}, b_{j\cdot k})$, fetched at step S411, the process proceeds to step S413, thereat the flag is set ON, and the process is completed.

[0116] As described above, in this example, the color gamut is easily checked by using the color gamut data table in Fig. 28.

[0117] According to the embodiment as described above, it is possible to realize a function of checking a color gamut easily in a color management system and the like. Accordingly, color reproduction quality can be improved as well as a time to check the color reproduction area and an image processing time can be shortened.

[0118] The present invention can be applied to a system constituted by a plurality of devices, or to an apparatus comprising a single device. Furthermore, the invention is applicable also to a case where the object of the invention is attained by supplying a program to a system or apparatus.

[0119] It should be noted that a color space representing a color gamut is not limited to the L* a* b* color space, and the standard RGB color space, such as NTSC, and the like can be used instead.

[0120] Further, if a value of ΔL* is not the minimum, an adding method can be added to the aforesaid method of checking a color gamut.

[0121] Furthermore, the present invention is not limited to an apparatus which displays the checked result of a color gamut, and can be applicable to any kinds of apparatus performing color reproduction on the basis of the checked result of the color gamut.

[0122] According to the present invention as described above, it is possible to provide an image processing method in which a color reproduction device converts image data that enable to reproduce a more faithful image to the input image.

[0123] Further, in a color management system and the like, a color gamut for realizing a color gamut checking function can be easily confirmed, thereby quality of determining color gamut can be improved as well as a color reproduction processing time can be shortened.

Claims

1. A color image processing method comprising the steps of:

    storing a color gamut table representing the colors which can be reproduced by a color reproduction device arranged to produce a color reproduction of color image data input to the device; and
    determining whether a color can be reproduced by the color reproduction device by using the stored color gamut table, **characterised by** carrying out the storing step by storing the color gamut table so that each address within the table is associated with a respective different color of the color input image data and the data stored at each address has either a first value indicating that the associated color falls within the color gamut or a second value indicating that the associated color falls outside the color gamut and carrying out the determining step by using the color image data to be input to the color reproduction device to address the color gamut table to determine whether the first or second value is stored at the address corresponding to that color and so to determine whether or not the color represented by the color data falls within the color gamut of the color reproduction device.

2. A method according to claim 1, wherein if it is determined in said determination step that the color image data is within the color gamut, a flag is turned on, and if it is determined in said determination step that the color image data is outside the color gamut, the flag is turned off.

3. A method according to claim 1, further comprising a step of interpolating data stored in a color gamut table.

4. A color image processing apparatus comprising:

    storage means (27) storing a color gamut table representing the colors which can be reproduced by a color reproduction device arranged to produce a color reproduction of color image data input to the device; and
    determination means (28) for determining whether a color can be reproduced by the color reproduction device by using the stored color gamut table, **characterised in that** each address within the color gamut table is associated with a respective different color of the color input image data and the data stored at each address has either a first value indicating that the associated color falls within the color gamut or a second value indicating that the associated color falls outside the color gamut and in that the determination means (28) is arranged to determine whether a color can be reproduced by the color reproduction device by using the color image data to be input to the color reproduction device to address the color gamut table to determine whether the first or second value is stored at the address corresponding to that color and so to determine whether or not the color represented by the color data falls within the color gamut of the color reproduction device.

5. An apparatus according to claim 4, wherein said determination means (28) is arranged to turn on a flag when said determination means (28) determines that the color image data is within the color gamut, and to turn off the flag when said determination means determines that the color image data is outside the color gamut.

6. An apparatus according to claim 4, further comprising an interpolation means for interpolating data stored in a color gamut table.

7. A computer program product comprising processor-implementable instructions for causing a processor

to carry out a method in accordance with claim 1 or 2.

8. A computer storage medium storing a computer program product in accordance with claim 7.

Patentansprüche

1. Farbbildverarbeitungsverfahren mit den Schritten

   Speichern einer Farbtonumfangtabelle, die die Farben darstellt, die durch eine Farbwiedergabeeinrichtung wiedergegeben werden können, die zur Erzeugung einer Farbwiedergabe von in die Einrichtung eingegebenen Farbbilddaten eingerichtet ist, und
   Bestimmen, ob eine Farbe durch die Farbwiedergabeeinrichtung wiedergegeben werden kann, indem die gespeicherte Farbtonumfangtabelle verwendet wird,

   dadurch gekennzeichnet, dass
   der Speicherschritt durch Speicherung der Farbtonumfangtabelle derart ausgeführt wird, dass jede Adresse in der Tabelle mit einer jeweils unterschiedlichen Farbe der Farbeingabebilddaten assoziiert wird, und die an jeder Adresse gespeicherten Daten entweder einen ersten Wert haben, der anzeigt, dass die assoziierte Farbe in den Farbtonumfang fällt, oder einen zweiten Wert haben, der anzeigt, das die assoziierte Farbe außerhalb des Farbtonumfangs liegt, und
   der Bestimmungsschritt durch Verwendung der in die Farbewiedergabeeinrichtung einzugebenden Farbbilddaten zur Adressierung der Farbtonumfangtabelle ausgeführt wird, um zu bestimmen, ob der erste oder der zweite Wert an der dieser Farbe entsprechenden Adresse gespeichert ist, um so zu bestimmen, ob die durch die Farbdaten dargestellte Farbe in den Farbtonumfang der Farbwiedergabeeinrichtung fällt oder nicht.

2. Verfahren nach Anspruch 1, wobei, wenn in dem Bestimmungsschritt bestimmt wird, dass die Farbbilddaten in dem Farbtonumfang liegen, eine Kennzeichnung eingeschaltet wird, und wenn in dem Bestimmungsschritt bestimmt wird, dass die Farbbilddaten außerhalb des Farbtonumfangs liegen, die Kennzeichnung ausgeschaltet wird.

3. Verfahren nach Anspruch 1, ferner mit dem Schritt der Interpolation von in einer Farbtonumfangtabelle gespeicherten Daten.

4. Farbbildverarbeitungsgerät mit

   einer Speichereinrichtung (27) zur Speicherung einer die Farben darstellenden Farbtonumfangtabelle, die durch eine Farbwiedergabeeinrichtung wiedergegeben werden können, die zur Erzeugung einer Farbwiedergabe von in die Einrichtung eingegebenen Farbbilddaten eingerichtete ist, und
   einer Bestimmungseinrichtung (28) zur Bestimmung, ob eine Farbe durch die Farbwiedergabeeinrichtung wiedergegeben werden kann, indem die gespeicherte Farbtonumfangtabelle verwendet wird,

   dadurch gekennzeichnet, dass
   jede Adresse in der Farbtonumfangtabelle mit einer jeweils unterschiedlichen Farbe der Farbeingabebilddaten assoziiert ist, und die an jeder Adresse gespeicherten Daten entweder einen ersten Wert haben, der anzeigt, dass die assoziierte Farbe in den Farbtonumfang fällt, oder einen zweiten Wert haben, der anzeigt, dass die assoziierte Farbe außerhalb des Farbtonumfangs liegt, und
   die Bestimmungseinrichtung (28) zur Bestimmung eingerichtet ist, ob eine Farbe durch die Farbwiedergabeeinrichtung wiedergegeben werden kann, indem die in die Farbwiedergabeeinrichtung einzugebenden Farbbilddaten zur Adressierung der Farbtonumfangtabelle zur Bestimmung verwendet werden, ob der erste oder der zweite Wert an der dieser Farbe entsprechenden Adresse gespeichert ist, und um so zu bestimmen, ob die durch die Farbdaten dargestellte Farbe in den Farbtonumfang der Farbwiedergabeeinrichtung fällt oder nicht.

5. Gerät nach Anspruch 4, wobei die Bestimmungseinrichtung (28) zum Einschalten einer Kennzeichnung eingerichtet ist, wenn die Bestimmungseinrichtung (28) bestimmt, dass die Farbbilddaten in dem Farbtonumfang liegen, und zum Ausschalten der Kennzeichnung eingerichtet ist, wenn die Bestimmungseinrichtung bestimmt, dass die Farbbilddaten außerhalb des Farbtonumfangs liegen.

6. Gerät nach Anspruch 4, ferner mit einer Interpolationseinrichtung zur Interpolation von in einer Farbtonumfangtabelle gespeicherten Daten.

7. Computerprogrammprodukt mit Prozessorimplementierbaren Befehlen, um einen Prozessor zur Ausrührung eines Verfahrens nach Anspruch 1 oder 2 zu veranlassen.

8. Computerspeicherträger, der ein Computerprogrammprodukt nach Anspruch 7 speichert.

Revendications

1. Procédé de traitement d'images en couleurs com-

prenant les étapes dans lesquelles :

on stocke une table de gamme de couleurs représentant les couleurs qui peuvent être reproduites par un dispositif de reproduction en couleurs agencé pour produire une reproduction en couleurs de données d'image en couleurs appliquées en entrée au dispositif ; et
on détermine si une couleur peut être reproduite par le dispositif de reproduction en couleurs en utilisant la table stockée de gamme de couleurs, **caractérisé par** l'exécution de l'étape de stockage en stockant la table de gamme de couleurs de manière que chaque adresse dans la table soit associée à une couleur différente respective des données d'image d'entrée en couleurs et que la donnée stockée à chaque adresse ait soit une première valeur indiquant que la couleur associée est comprise dans la gamme de couleurs, soit une seconde valeur indiquant que la couleur associée est en dehors de la gamme de couleurs, et l'exécution de l'étape de détermination en utilisant les données d'image en couleurs devant être appliquées en entrée au dispositif de reproduction en couleurs pour adresser la table de gamme de couleurs afin de déterminer si la première ou seconde valeur est stockée à l'adresse correspondant à cette couleur et de déterminer ainsi si la couleur représentée par la donnée de couleurs est comprise ou non dans la gamme de couleurs du dispositif de reproduction en couleurs.

2. Procédé selon la revendication 1, dans lequel, s'il est déterminé dans ladite étape de détermination que les données d'image en couleurs sont comprises dans la gamme de couleurs, un drapeau est présenté, et s'il est déterminé dans ladite étape de détermination que les données d'image en couleurs sont en dehors de la gamme de couleurs, le drapeau est effacé.

3. Procédé selon la revendication 1, comprenant en outre une étape d'interpolation de données stockées dans une table de gamme de couleurs.

4. Appareil de traitement d'images en couleurs comportant :

un moyen de stockage (27) stockant une table de gamme de couleurs représentant les couleurs qui peuvent être reproduites par un dispositif de reproduction en couleurs agencé pour produire une reproduction en couleurs de données d'image en couleurs appliquées en entrée au dispositif ; et
un moyen de détermination (28) destiné à déterminer si une couleur peut être reproduite par le dispositif de reproduction en couleurs en utilisant la table de gamme de couleurs stockée, **caractérisé en ce que** chaque adresse dans la table de gamme de couleurs est associée à une couleur différente respective des données d'image d'entrée de couleurs et la donnée stockée à chaque adresse possède soit une première valeur indiquant que la couleur associée est comprise dans la gamme de couleurs, soit une seconde valeur indiquant que la couleur associée est en dehors de la gamme de couleurs, et en ce que le moyen de détermination (28) est agencé pour déterminer si une couleur peut être reproduite par le dispositif de reproduction en couleurs en utilisant les données d'image en couleurs devant être appliquées en entrée au dispositif de reproduction en couleurs pour adresser la table de gamme de couleurs afin de déterminer si la première ou seconde valeur est stockée à l'adresse correspondant à cette couleur et déterminer ainsi si la couleur représentée par la donnée de couleurs est comprise ou non dans la gamme de couleurs du dispositif de reproduction en couleurs.

5. Appareil selon la revendication 4, dans lequel ledit moyen (28) de détermination est agencé pour présenter un drapeau lorsque ledit moyen de détermination (28) détermine que la donnée d'image en couleurs est comprise dans la gamme de couleurs, et pour effacer le drapeau lorsque ledit moyen de détermination détermine que la donnée d'image en couleurs est en dehors de la gamme de couleurs.

6. Appareil selon la revendication 4, comportant en outre un moyen d'interpolation destiné à interpoler des données stockées dans une table de gamme de couleurs.

7. Produit à programme d'ordinateur comprenant des instructions pouvant être exécutées par un processeur pour amener un processeur à mettre en oeuvre un procédé selon la revendication 1 ou 2.

8. Support de stockage pour ordinateur stockant un produit à programme d'ordinateur selon la revendication 7.

FIG. 1

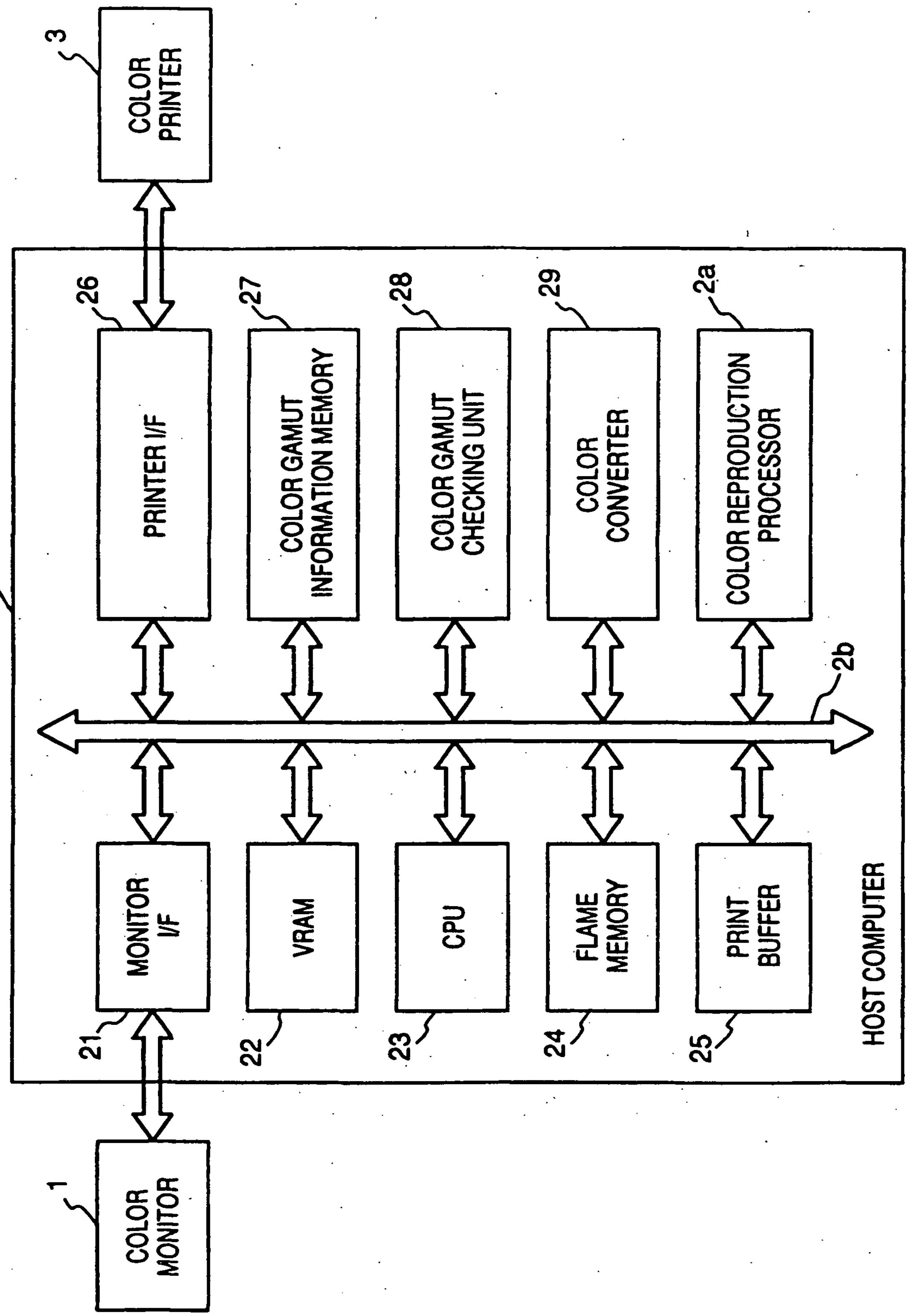

COLOR PRINTER
3
2
26
PRINTER I/F
27
COLOR GAMUT INFORMATION MEMORY
28
COLOR GAMUT CHECKING UNIT
29
COLOR CONVERTER
2a
COLOR REPRODUCTION PROCESSOR
2b
MONITOR I/F
21
VRAM
22
CPU
23
FLAME MEMORY
24
PRINT BUFFER
25
HOST COMPUTER
COLOR MONITOR
1

FIG. 2

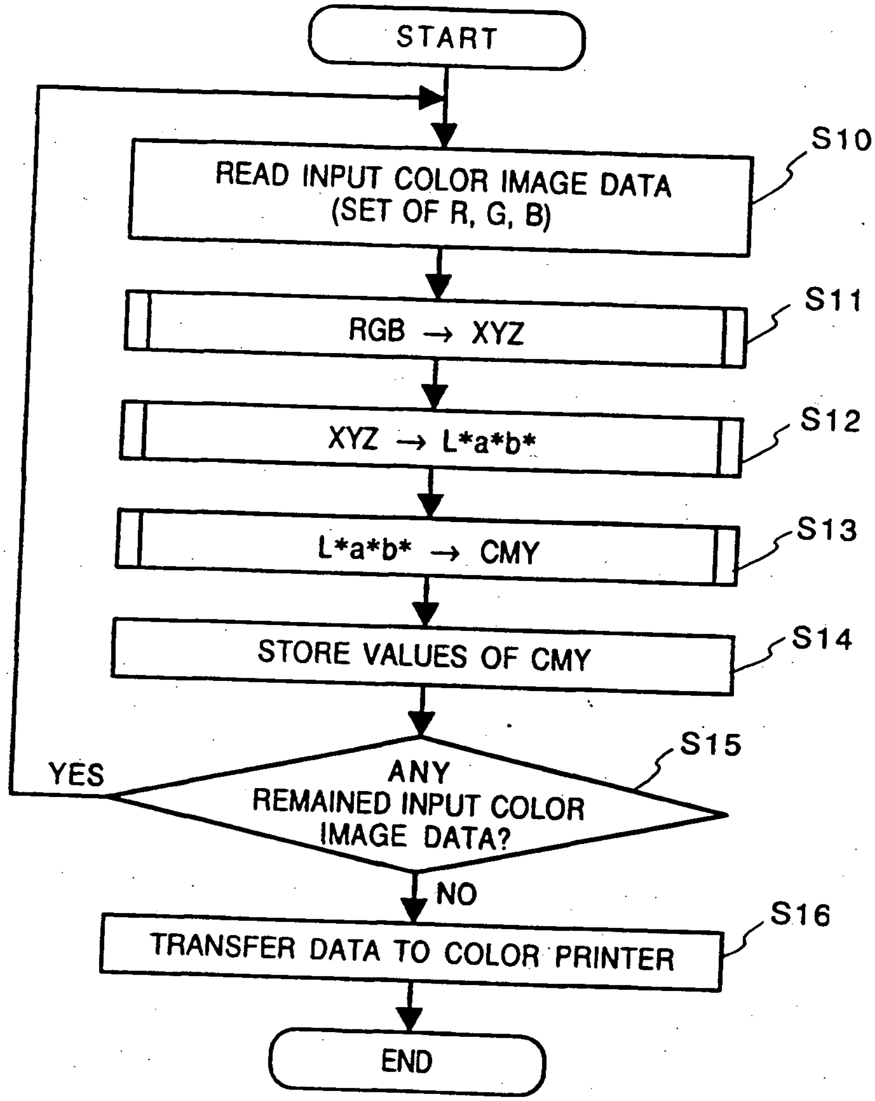


FIG. 3

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

FIG. 4

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

FIG. 5

$$\left\{ \begin{array}{l} L^* = \begin{cases} 116\,(Y/Y_n)^{\frac{1}{3}} - 16 & (Y/Y_n > 0.008856) \\ 903.3\,(Y/Y_n) & (Y/Y_n < 0.008856) \end{cases} \\[2ex] a^* = \begin{cases} 500\,[(X/X_n)^{\frac{1}{3}} - (Y/Y_n)^{\frac{1}{3}}] & \begin{pmatrix} X/X_n > 0.008856 \\ Y/Y_n > 0.008856 \end{pmatrix} \\ 500\,[f\,(X/X_n) - f\,(Y/Y_n)\,] & \begin{pmatrix} X/X_n < 0.008856 \\ Y/Y_n < 0.008856 \end{pmatrix} \end{cases} \\[2ex] b^* = \begin{cases} 200\,[(Y/Y_n)^{\frac{1}{3}} - (Z/Z_n)^{\frac{1}{3}}] & (Z/Z_n > 0.008856) \\ 200\,[f\,(Y/Y_n) - f\,(Z/Z_n)\,] & (Z/Z_n < 0.008856) \end{cases} \\[2ex] \qquad f\,(X/X_n) = 7.787\,(X/X_n) + 16/116 \\ \qquad f\,(Y/Y_n) = 7.787\,(Y/Y_n) + 16/116 \\ \qquad f\,(Z/Z_n) = 7.787\,(Z/Z_n) + 16/116 \end{array} \right.$$

FIG. 6

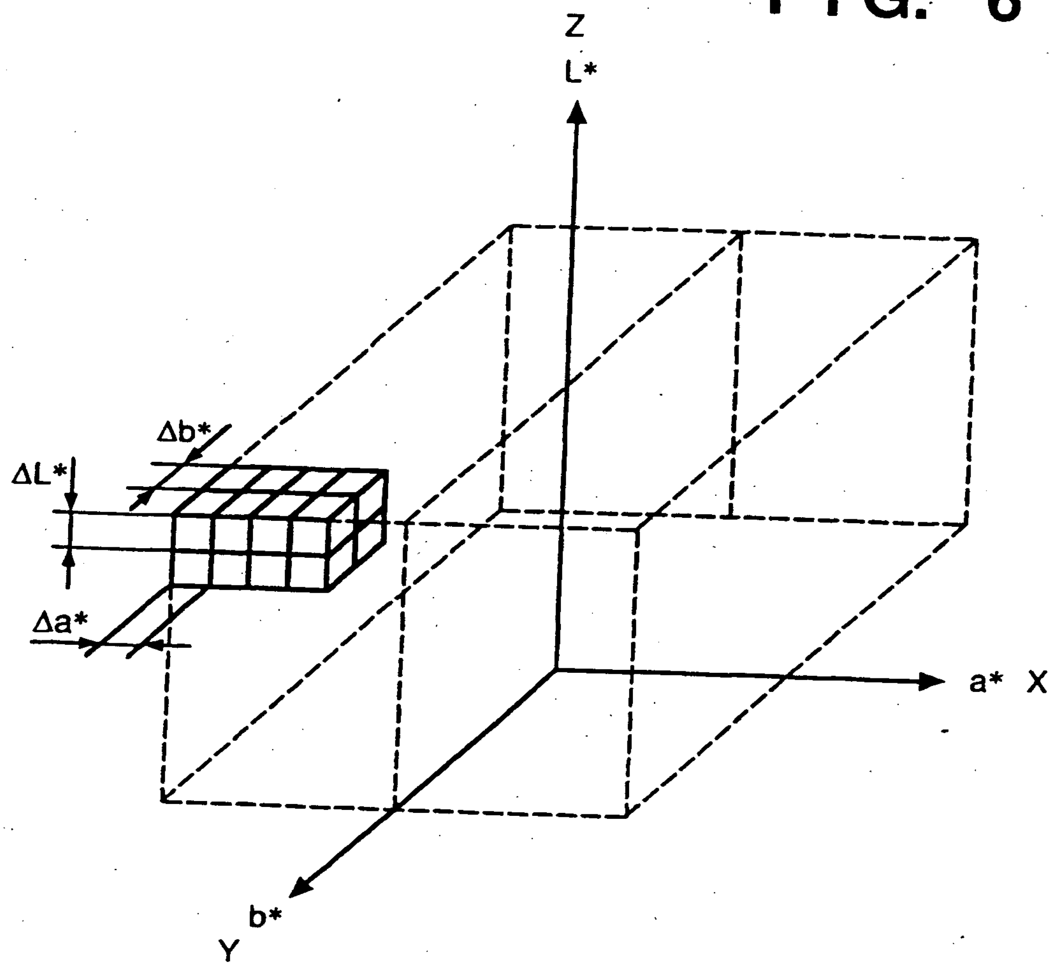
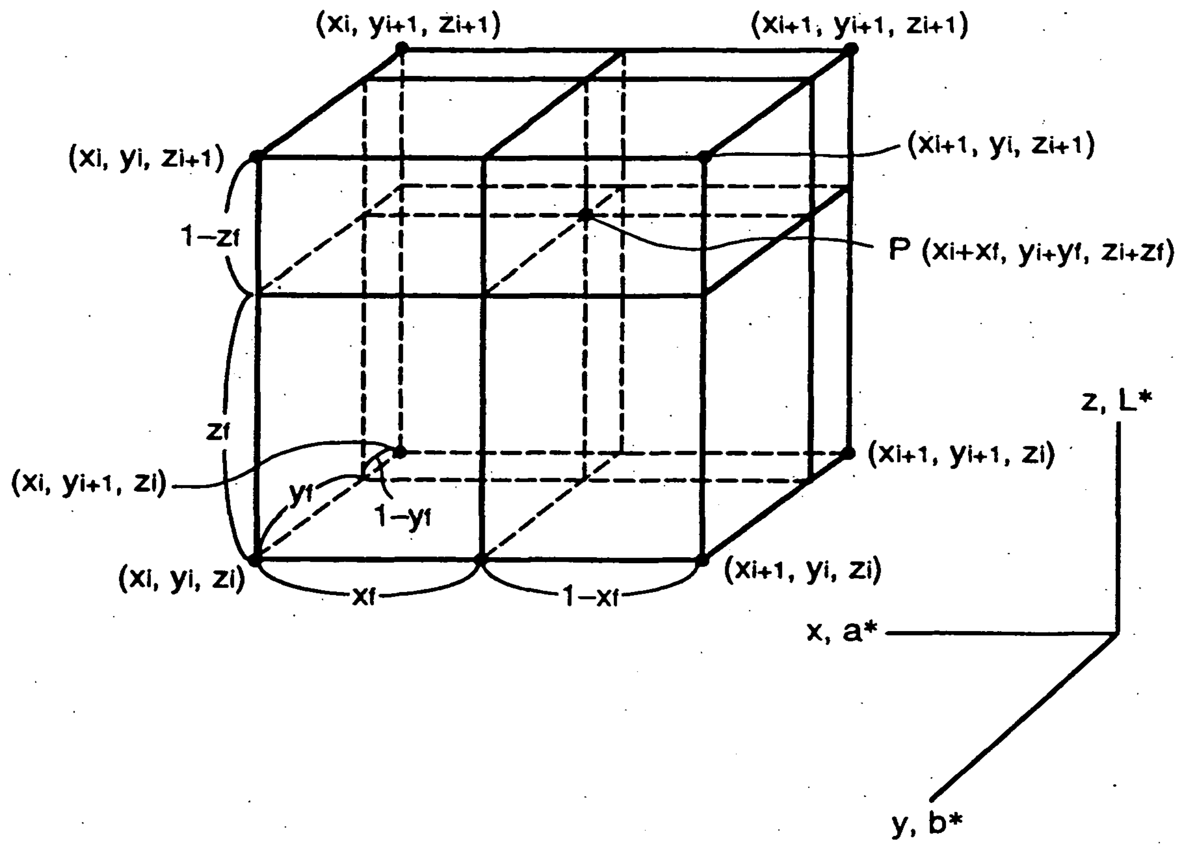


FIG. 7



(xi, yi+1, zi+1)
(xi+1, yi+1, zi+1)
(xi, yi, zi+1)
(xi+1, yi, zi+1)
1−zf
P (xi+xf, yi+yf, zi+zf)
zf
(xi, yi+1, zi)
(xi+1, yi+1, zi)
yf
1−yf
(xi, yi, zi)
(xi+1, yi, zi)
xf
1−xf
z, L*
x, a*
y, b*

FIG. 8

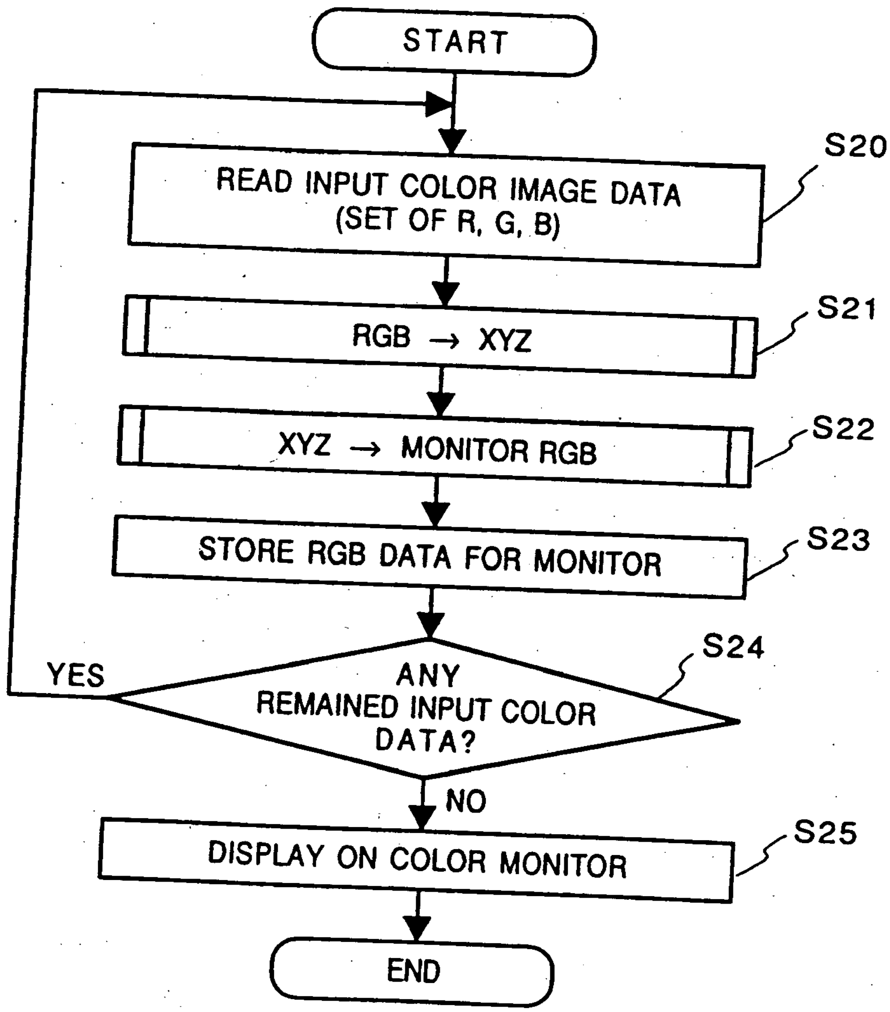

START
READ INPUT COLOR IMAGE DATA
(SET OF R, G, B)
S20
RGB → XYZ
S21
XYZ → MONITOR RGB
S22
STORE RGB DATA FOR MONITOR
S23
ANY
REMAINED INPUT COLOR
DATA?
S24
YES
NO
DISPLAY ON COLOR MONITOR
S25
END

FIG. 9

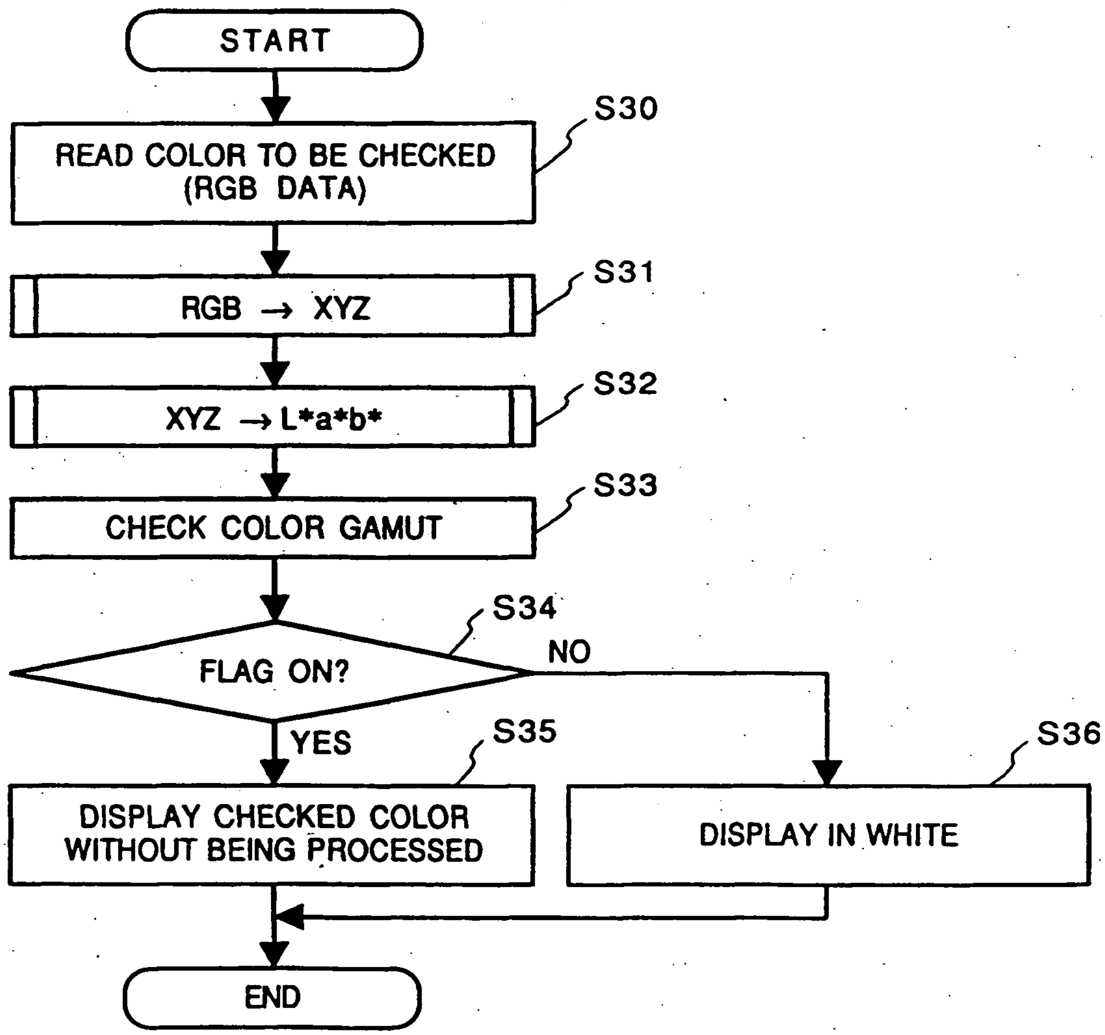

START
S30
READ COLOR TO BE CHECKED
(RGB DATA)
S31
RGB → XYZ
S32
XYZ → L*a*b*
S33
CHECK COLOR GAMUT
S34
FLAG ON?
NO
YES
S35
DISPLAY CHECKED COLOR
WITHOUT BEING PROCESSED
S36
DISPLAY IN WHITE
END

FIG. 10

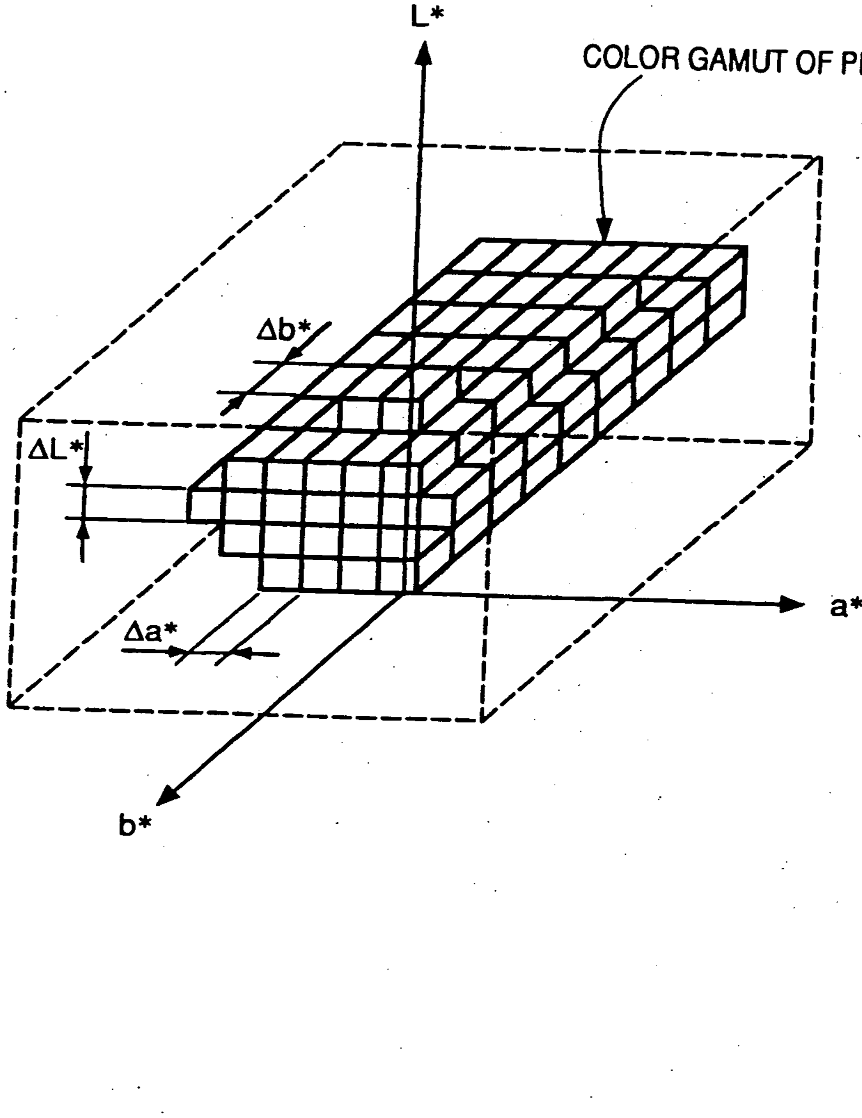


FIG. 11

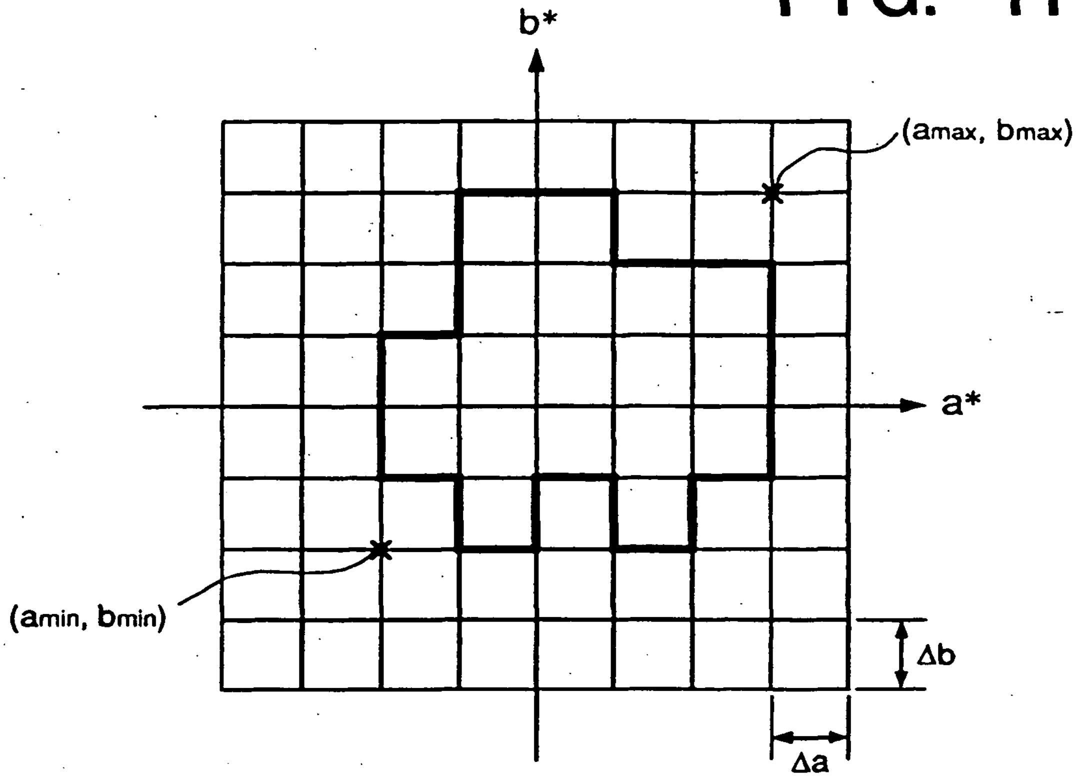


FIG. 12

L1 (L*min) →	a1min	b1min	a1max	b1max
L2	a2min	b2min	a2max	b2max
		⋮		
Lm (L*max)	ammin	bmmin	ammax	bmmax

FIG. 13

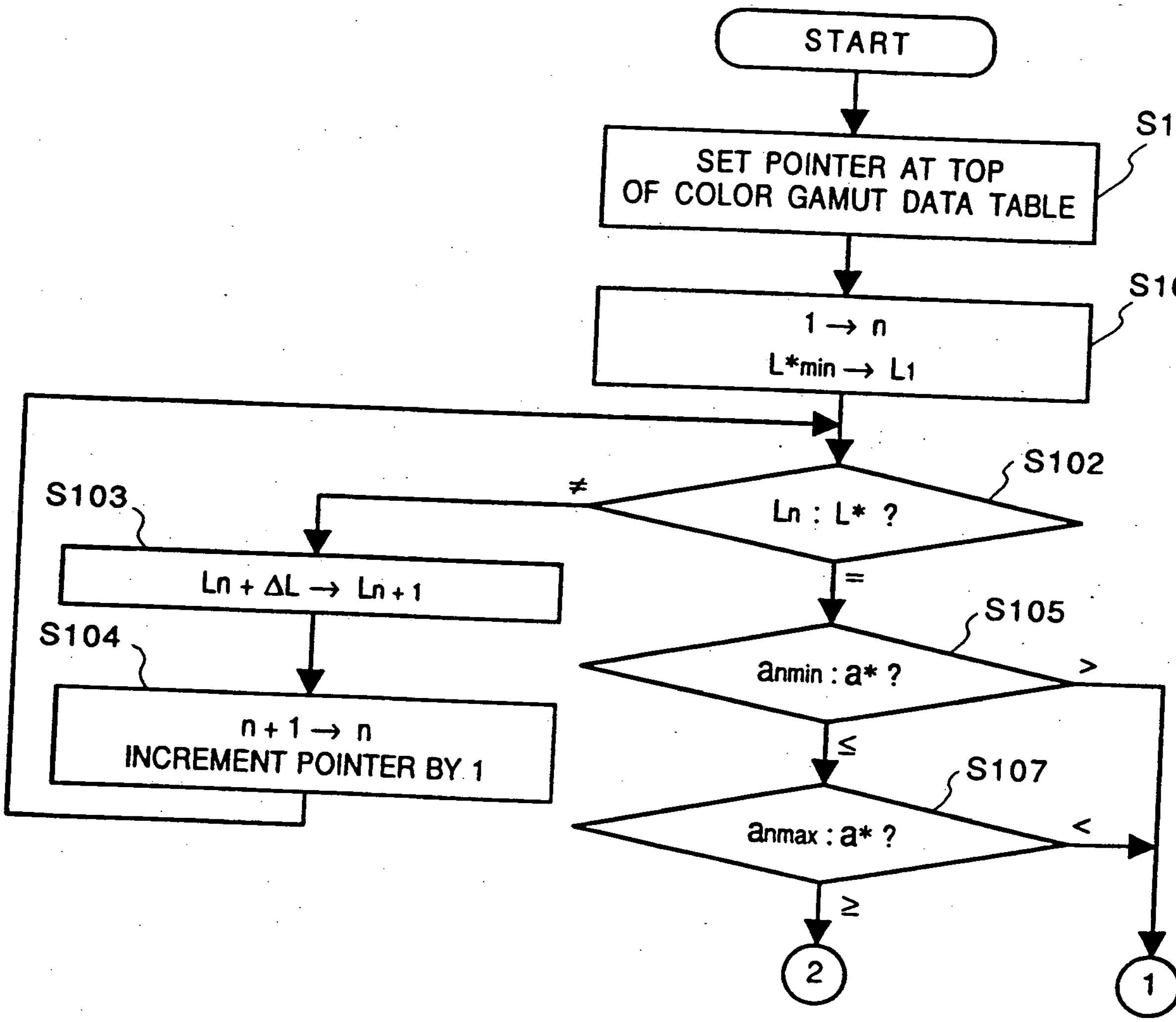

START
S100
SET POINTER AT TOP OF COLOR GAMUT DATA TABLE
S101
1 → n
L*min → L1
S102
Ln : L* ?
≠
=
S103
Ln + ΔL → Ln+1
S104
n + 1 → n
INCREMENT POINTER BY 1
S105
anmin : a* ?
>
≤
S107
anmax : a* ?
<
≥
2
1

FIG. 14

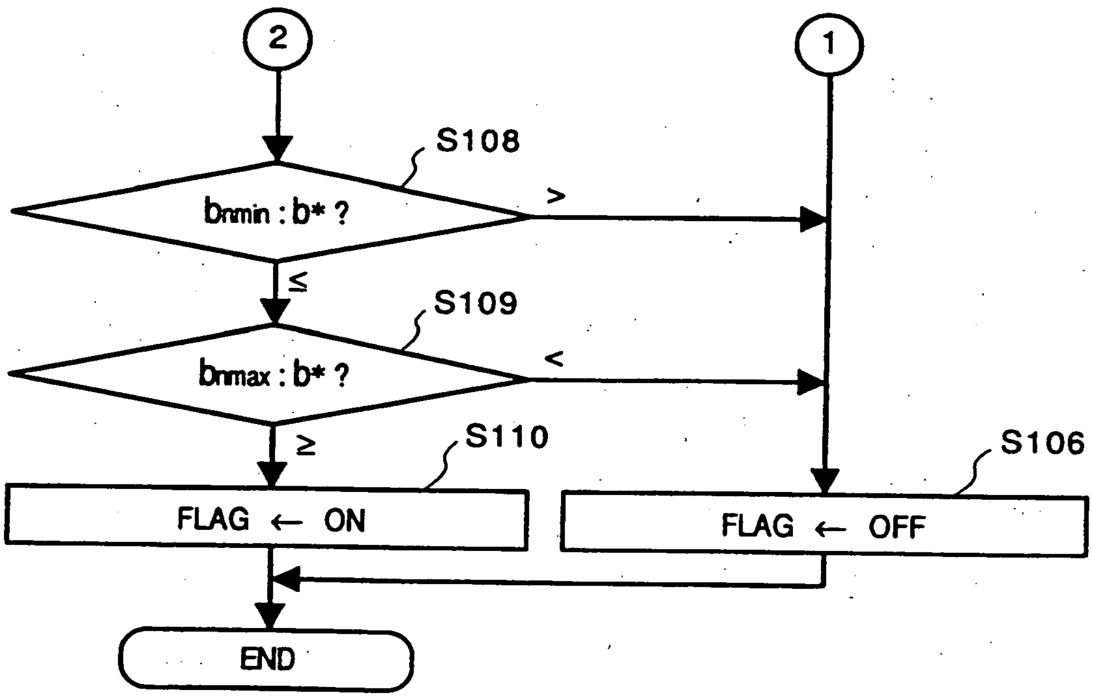


FIG. 15

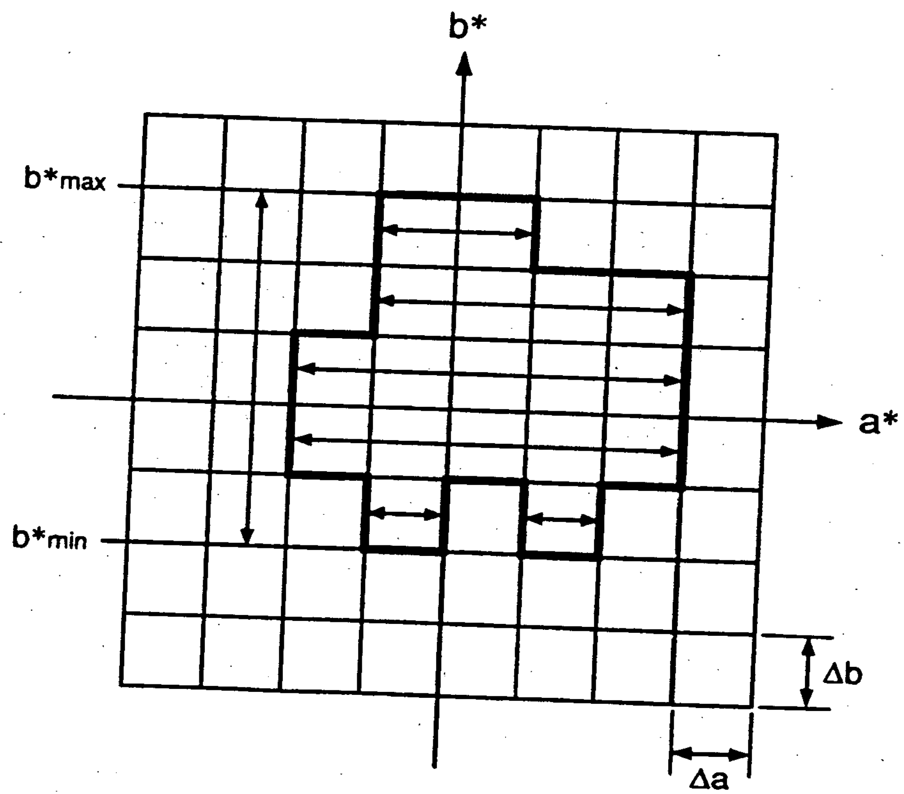


FIG. 16

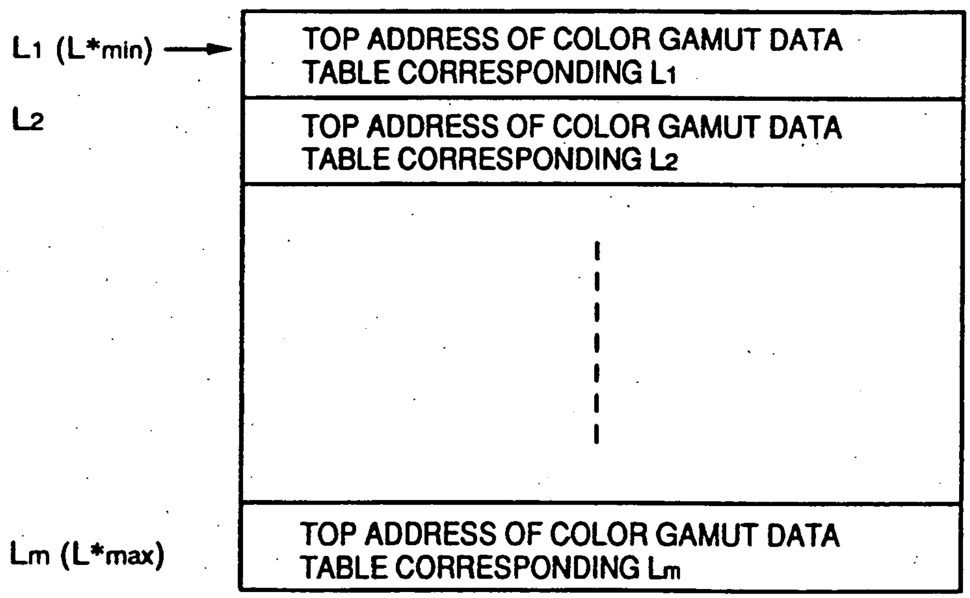

L_1 (L^*min)
TOP ADDRESS OF COLOR GAMUT DATA TABLE CORRESPONDING L_1
L_2
TOP ADDRESS OF COLOR GAMUT DATA TABLE CORRESPONDING L_2
L_m (L^*max)
TOP ADDRESS OF COLOR GAMUT DATA TABLE CORRESPONDING L_m

FIG. 17

b^*_{min} →	i1	a_{11min}	b_{11max}	a_{12min}	b_{12max}	- - -	a_{1i1min}	b_{1i1max}
	i2	a_{21min}	b_{21max}	a_{22min}	b_{22max}	- - -	a_{2i2min}	b_{2i2max}
	⋮			⋮				
b^*_{max}	in	a_{n1min}	b_{n1max}	a_{n2min}	b_{n2max}	- - -	a_{ninmin}	b_{ninmax}

FIG. 18

2	-1	0	1	2
1	-2	3		
1	-2	3		
1	-1	3		
1	-1	1		

FIG. 19

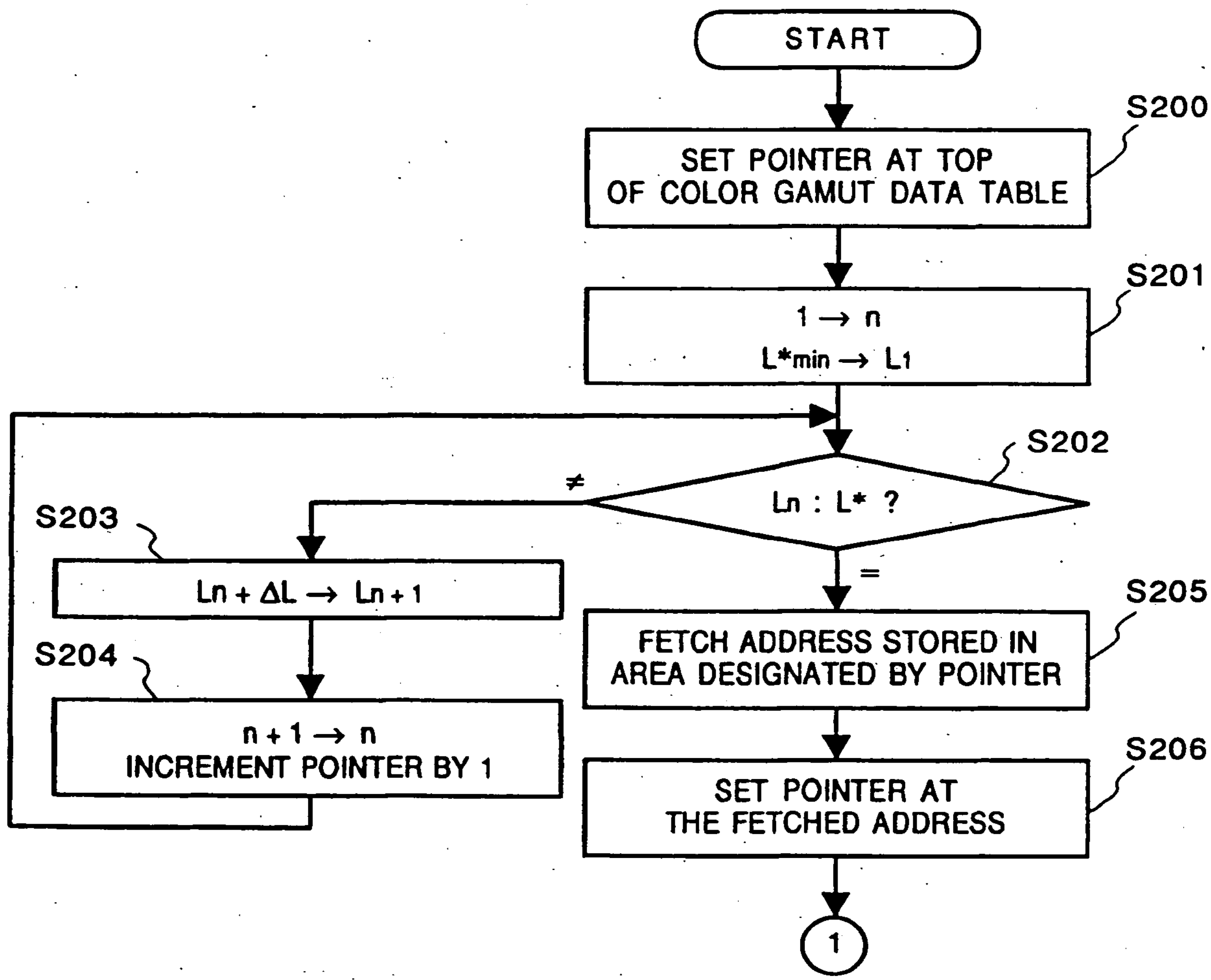

START
S200
SET POINTER AT TOP OF COLOR GAMUT DATA TABLE
S201
1 → n
L*min → L1
S202
Ln : L* ?
≠
=
S203
Ln + ΔL → Ln + 1
S204
n + 1 → n
INCREMENT POINTER BY 1
S205
FETCH ADDRESS STORED IN AREA DESIGNATED BY POINTER
S206
SET POINTER AT THE FETCHED ADDRESS
1

FIG. 20

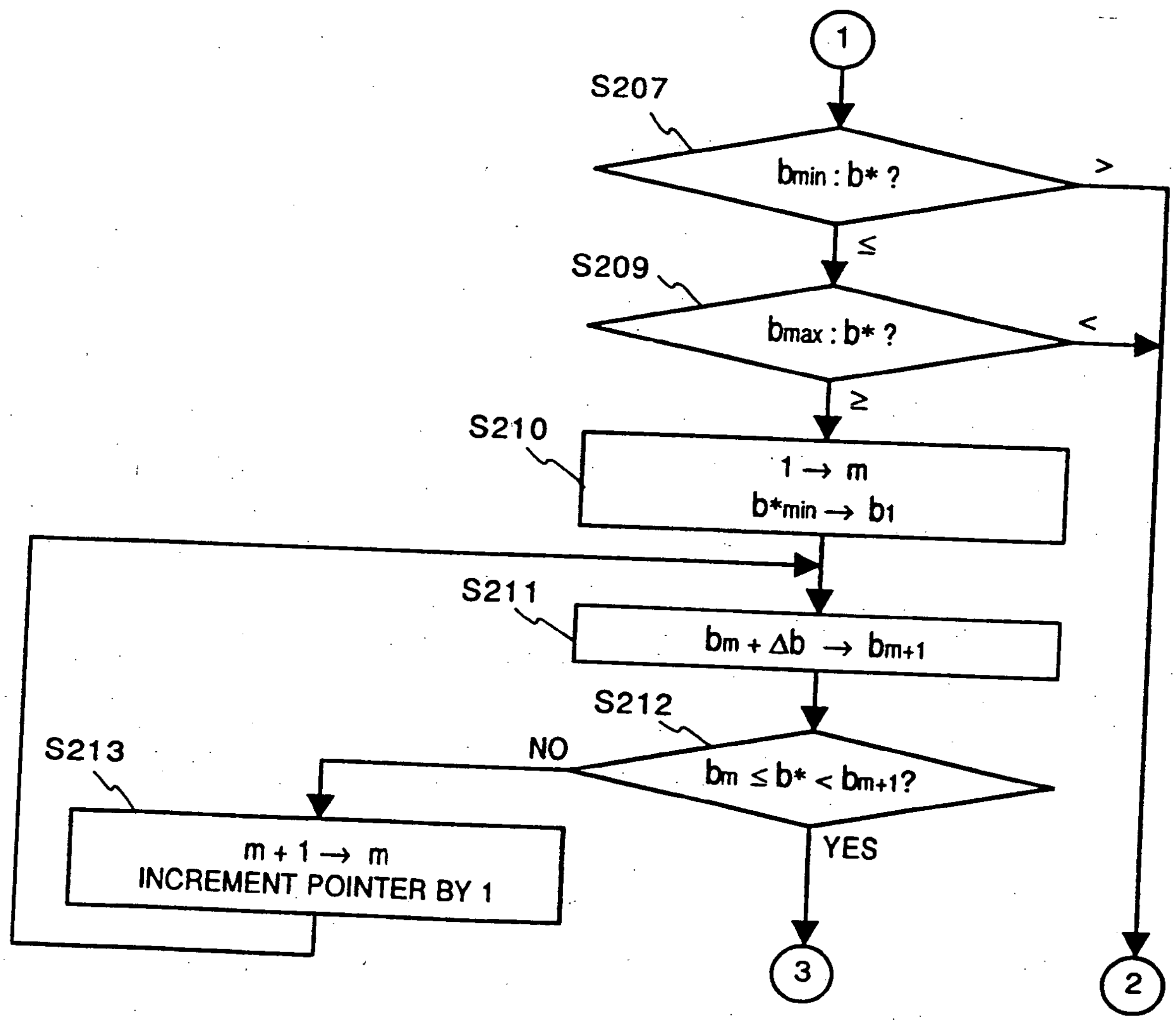

1
S207
bmin : b* ?
>
≤
S209
bmax : b* ?
<
≥
S210
1 → m
b*min → b1
S211
bm + Δb → bm+1
S212
bm ≤ b* < bm+1?
NO
YES
S213
m + 1 → m
INCREMENT POINTER BY 1
3
2

FIG. 21

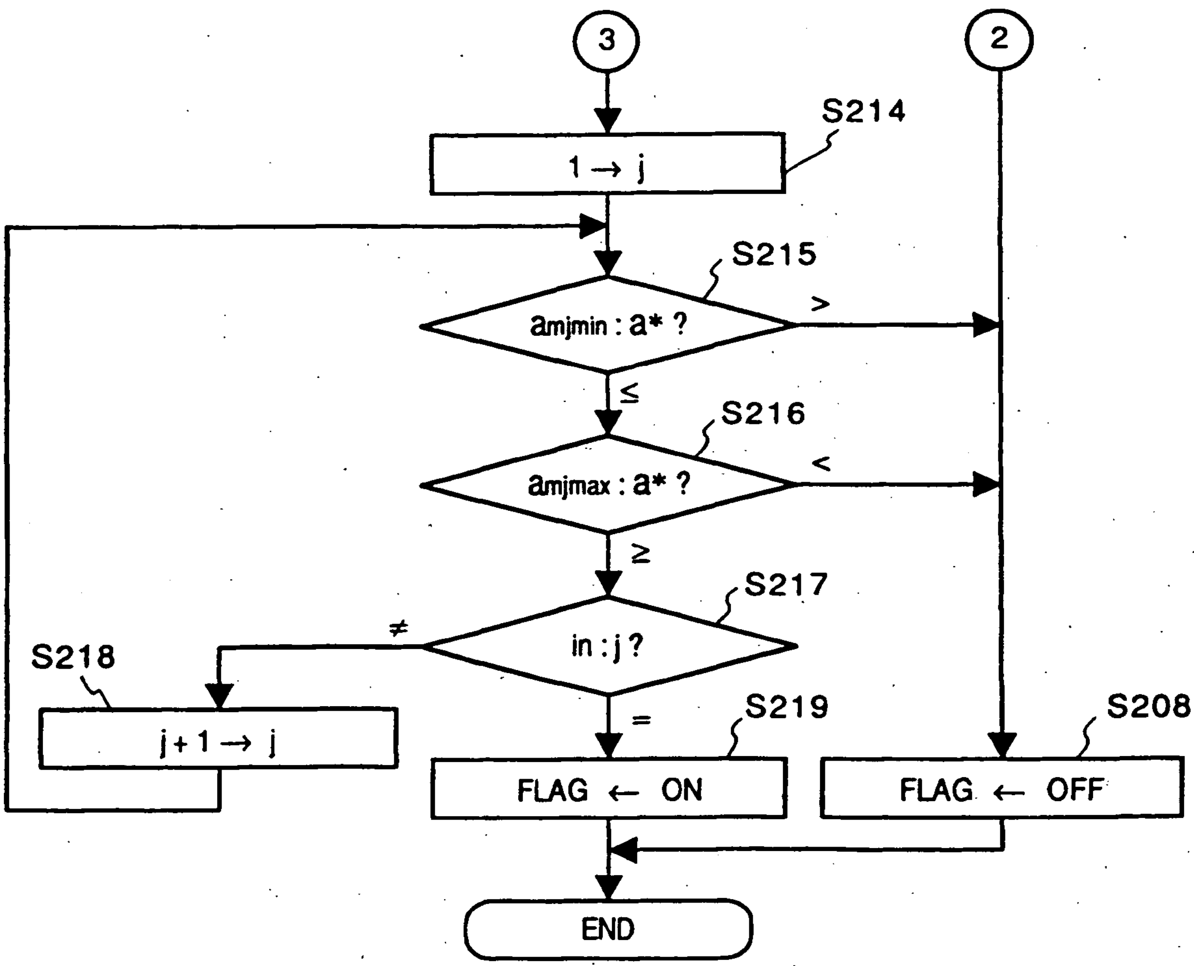

3
2
S214
1 → j
S215
amjmin : a* ?
>
≤
S216
amjmax : a* ?
<
≥
S217
in : j ?
≠
=
S218
j + 1 → j
S219
FLAG ← ON
S208
FLAG ← OFF
END

FIG. 22

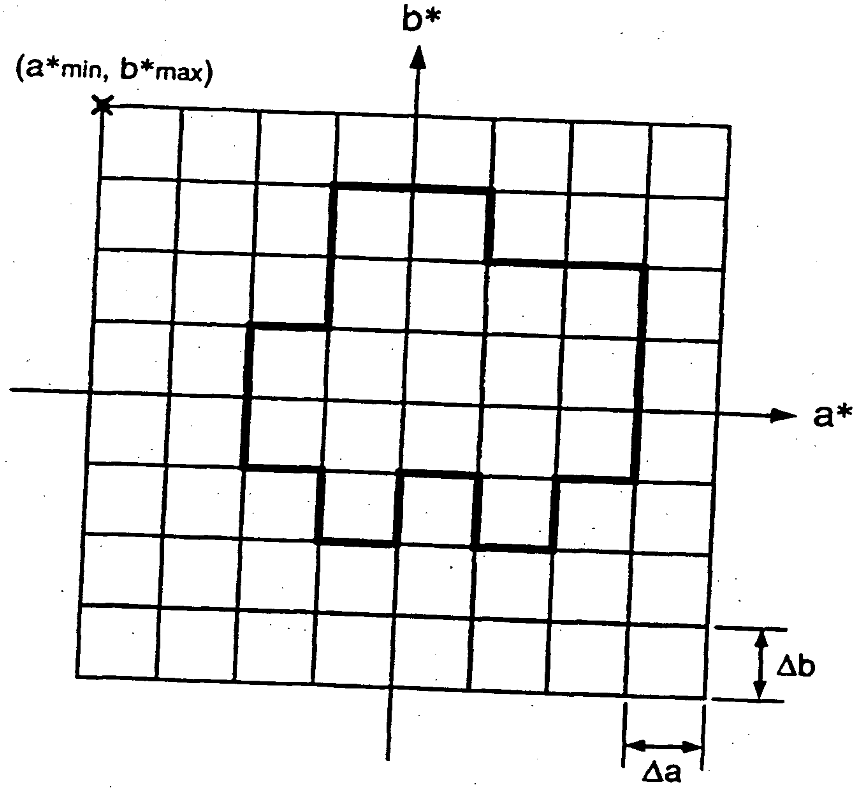


FIG. 23

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	0	1	1	1	1	1	0
0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIG. 24

L_1 (L^*min) →	TOP ADDRESS OF COLOR GAMUT DATA TABLE CORRESPONDING L_1
L_2	TOP ADDRESS OF COLOR GAMUT DATA TABLE CORRESPONDING L_2
	⋮
L_m (L^*max)	TOP ADDRESS OF COLORGAMUT DATA TABLE CORRESPONDING L_m

FIG. 25

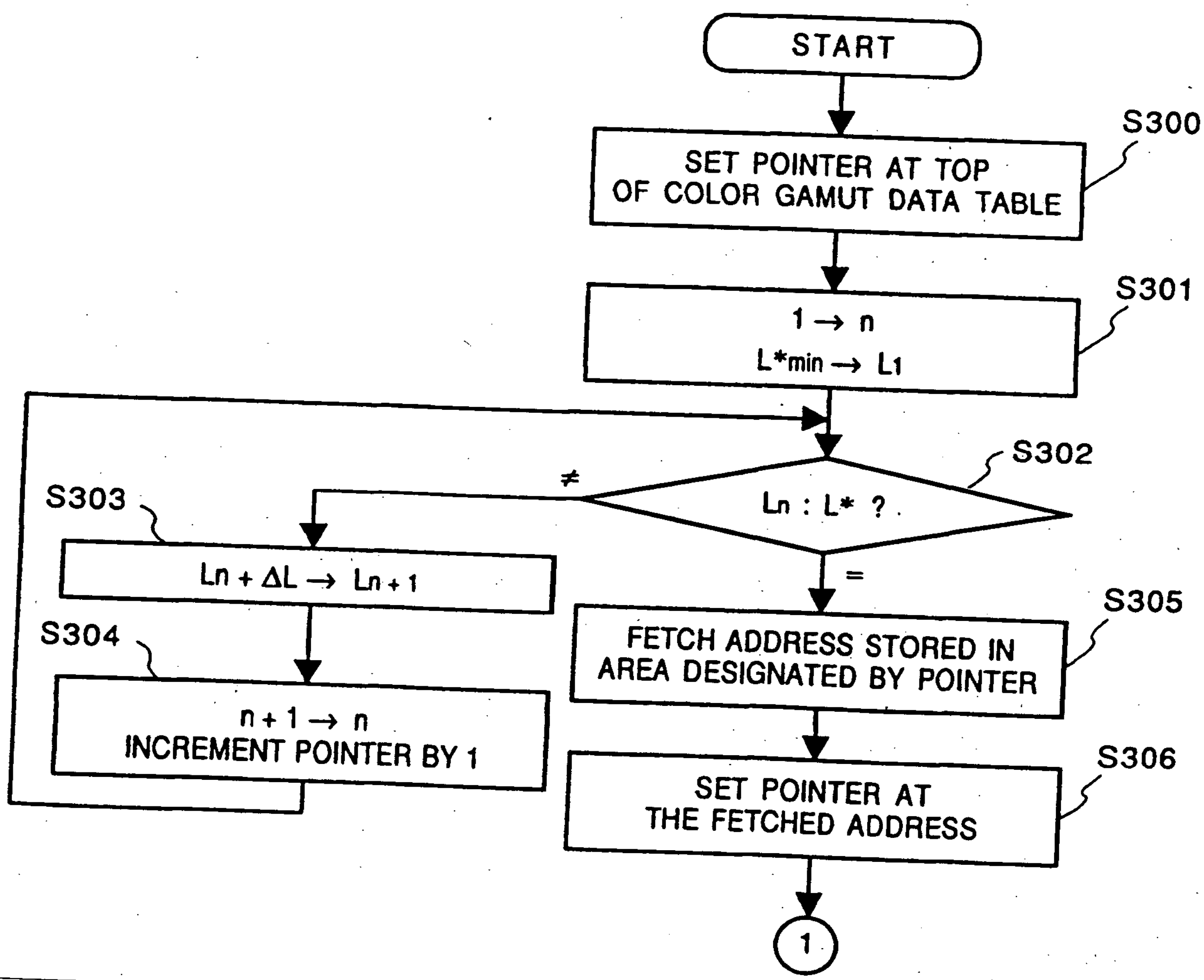

START
S300
SET POINTER AT TOP OF COLOR GAMUT DATA TABLE
S301
1 → n
L*min → L1
S302
Ln : L* ?
≠
=
S303
Ln + ΔL → Ln + 1
S304
n + 1 → n
INCREMENT POINTER BY 1
S305
FETCH ADDRESS STORED IN AREA DESIGNATED BY POINTER
S306
SET POINTER AT THE FETCHED ADDRESS
1

FIG. 26

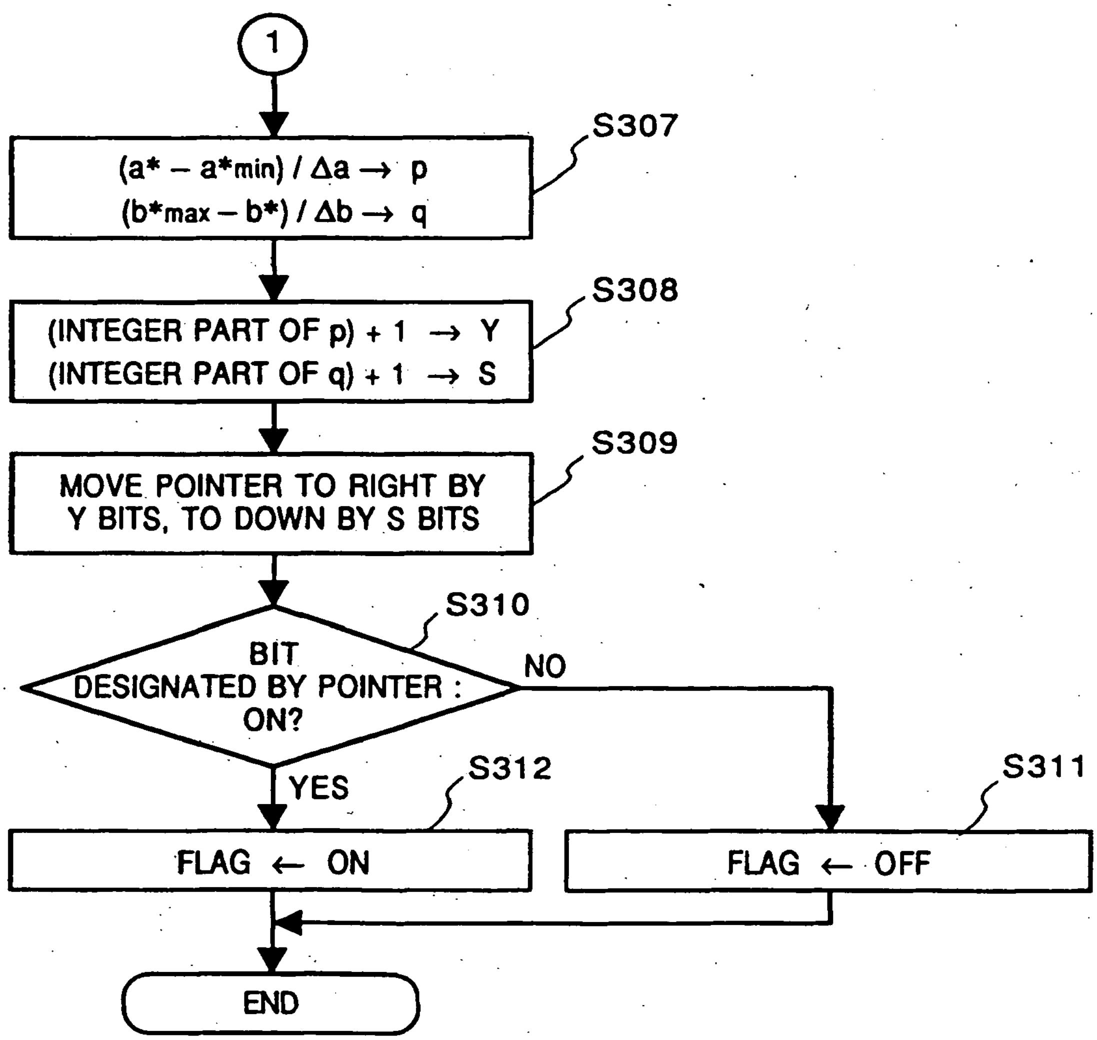

1
S307
(a* – a*min) / Δa → p
(b*max – b*) / Δb → q
S308
(INTEGER PART OF p) + 1 → Y
(INTEGER PART OF q) + 1 → S
S309
MOVE POINTER TO RIGHT BY
Y BITS, TO DOWN BY S BITS
S310
BIT
DESIGNATED BY POINTER :
ON?
NO
YES
S312
FLAG ← ON
S311
FLAG ← OFF
END

FIG. 27

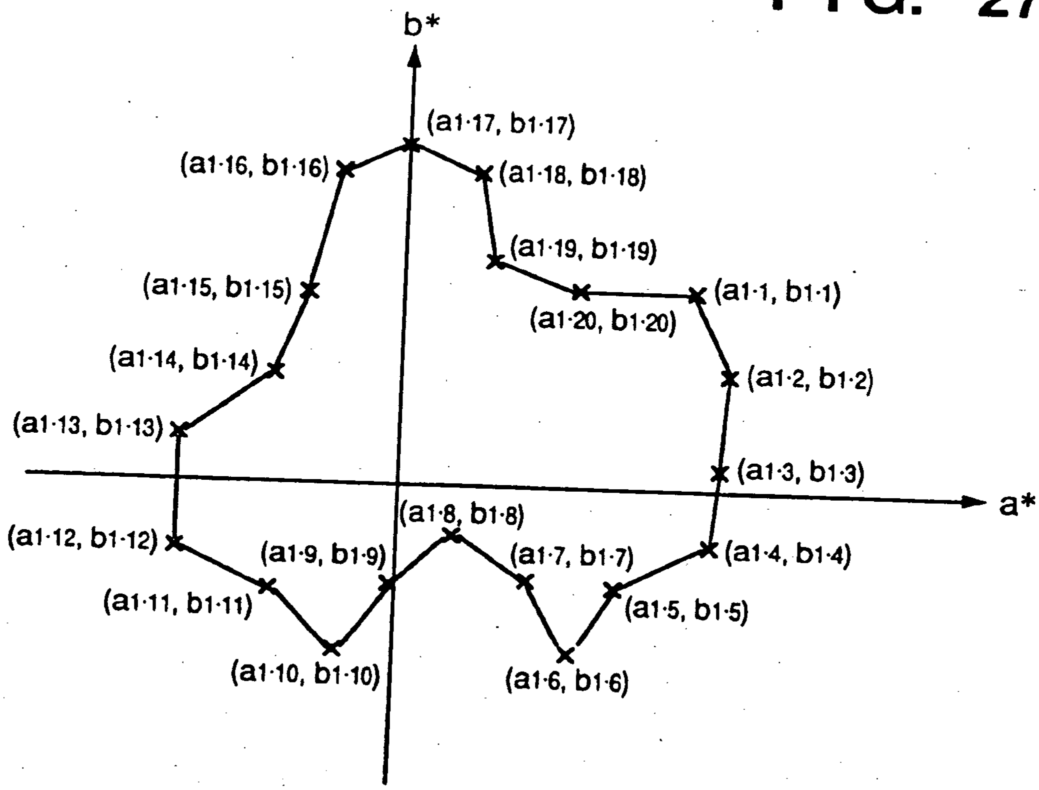


FIG. 28

L1 (L*min) →	i1	a1·1	b1·1	a1·2	b1·2	- - -	a1·i1	b1·i1
L2	i2	a2·1	b2·1	a2·2	b2·2	- - -	a2·i2	b2·i2
	⋮	⋮						
Lm (L*max)	in	am·1	bm·1	am·2	bm·2	- - -	am·in	bm·in

FIG. 29

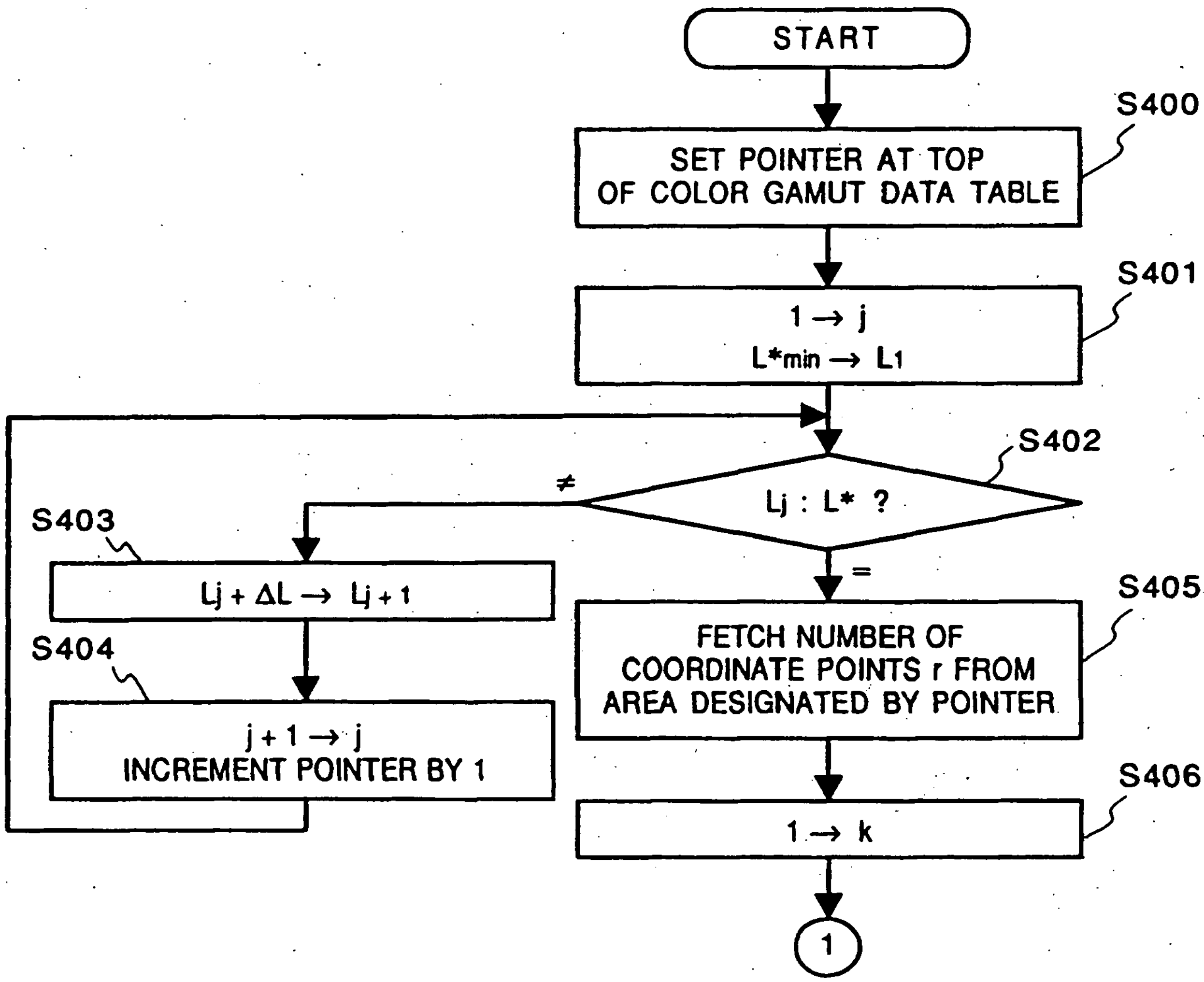

START
S400
SET POINTER AT TOP OF COLOR GAMUT DATA TABLE
S401
1 → j
L*min → L1
S402
Lj : L* ?
≠
=
S403
Lj + ΔL → Lj + 1
S404
j + 1 → j
INCREMENT POINTER BY 1
S405
FETCH NUMBER OF COORDINATE POINTS r FROM AREA DESIGNATED BY POINTER
S406
1 → k
1

FIG. 30

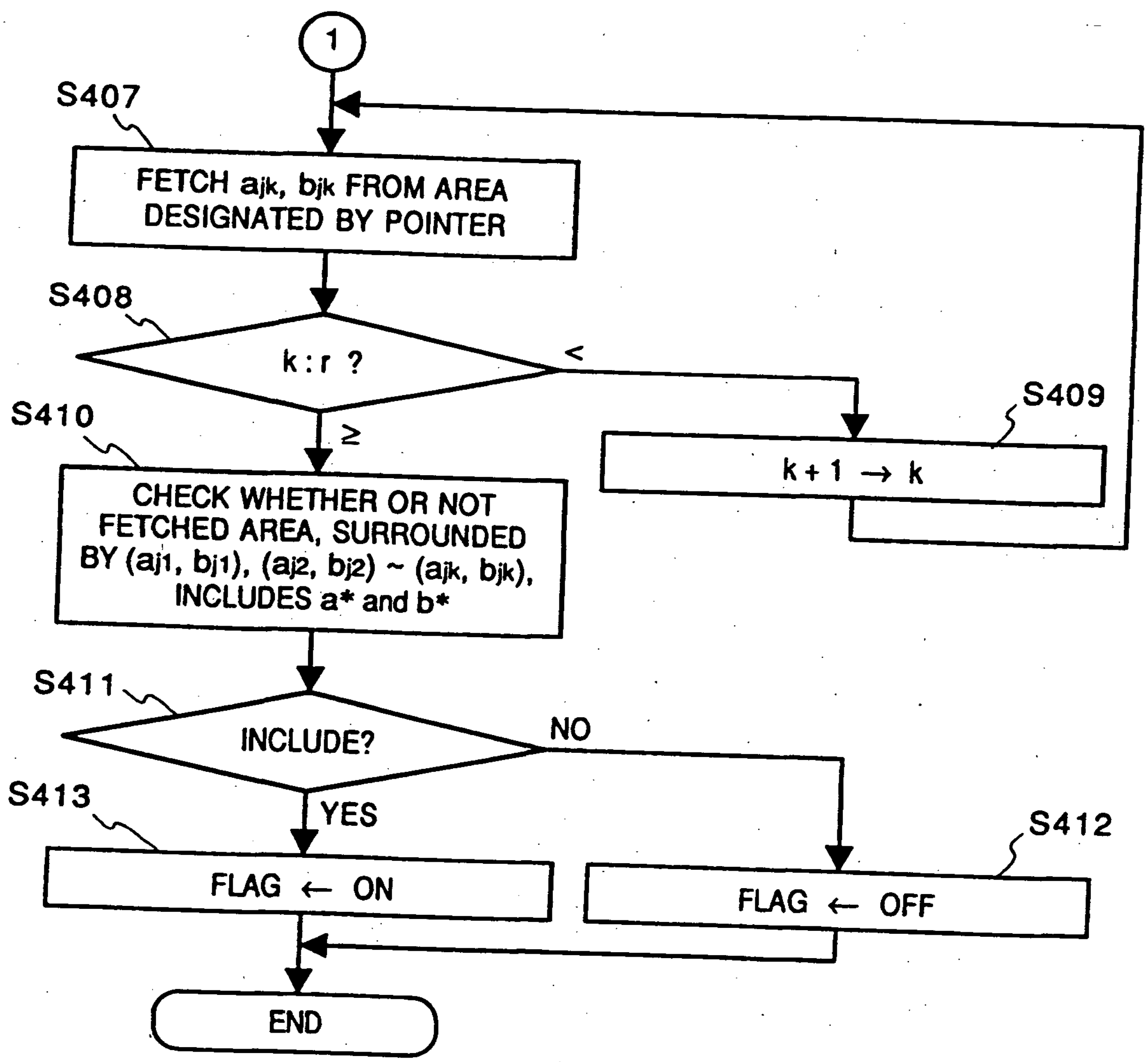

1
S407
FETCH ajk, bjk FROM AREA DESIGNATED BY POINTER
S408
k : r ?
<
≥
S409
k + 1 → k
S410
CHECK WHETHER OR NOT FETCHED AREA, SURROUNDED BY (aj1, bj1), (aj2, bj2) ~ (ajk, bjk), INCLUDES a* and b*
S411
INCLUDE?
NO
YES
S413
FLAG ← ON
S412
FLAG ← OFF
END

FIG. 31

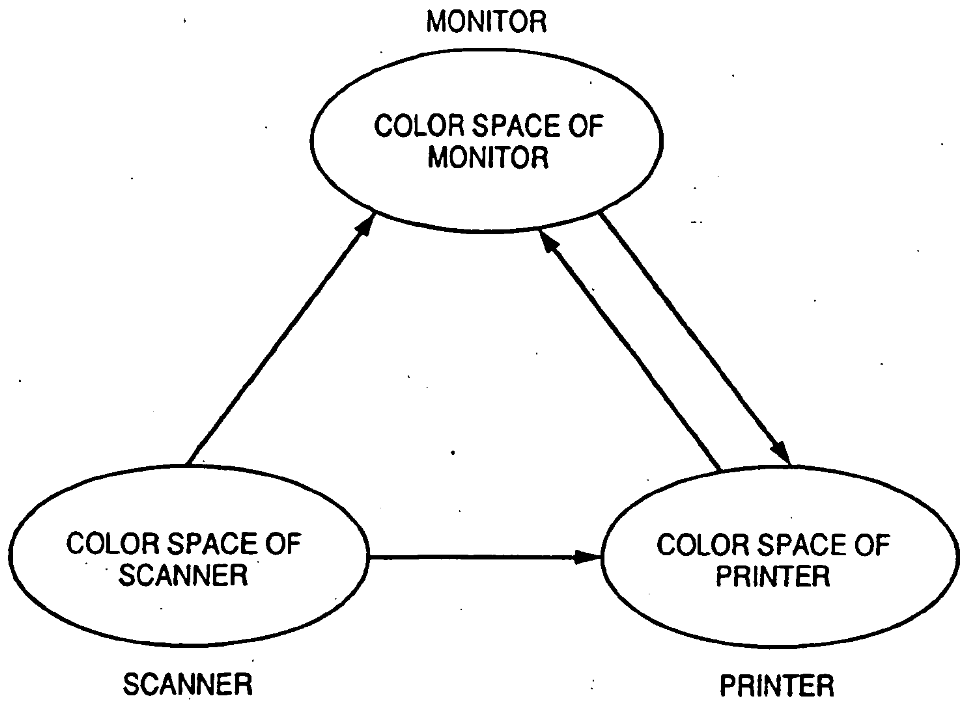

MONITOR
COLOR SPACE OF MONITOR
COLOR SPACE OF SCANNER
COLOR SPACE OF PRINTER
SCANNER
PRINTER

FIG. 32

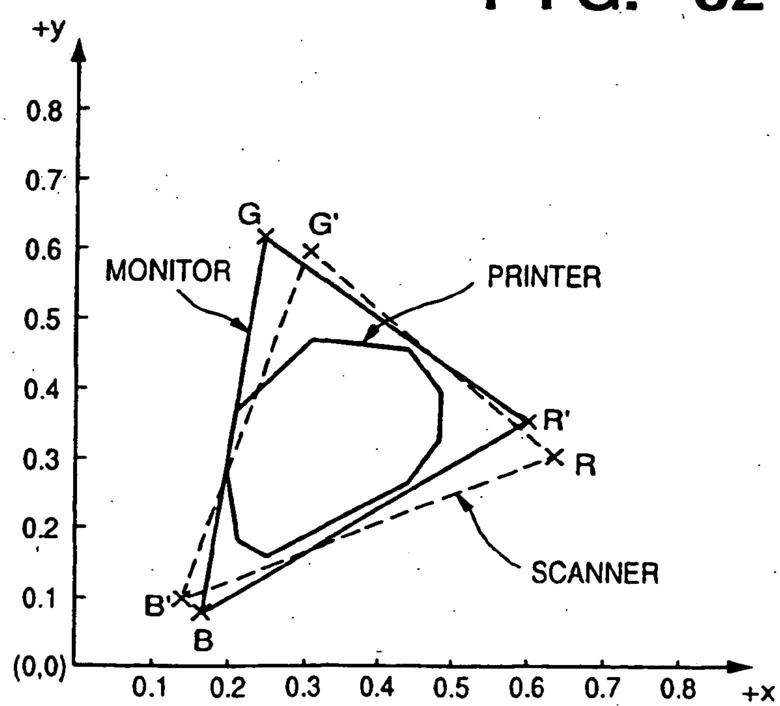

+y
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
(0,0)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
+x
G
G'
MONITOR
PRINTER
R'
R
SCANNER
B'
B

FIG. 33

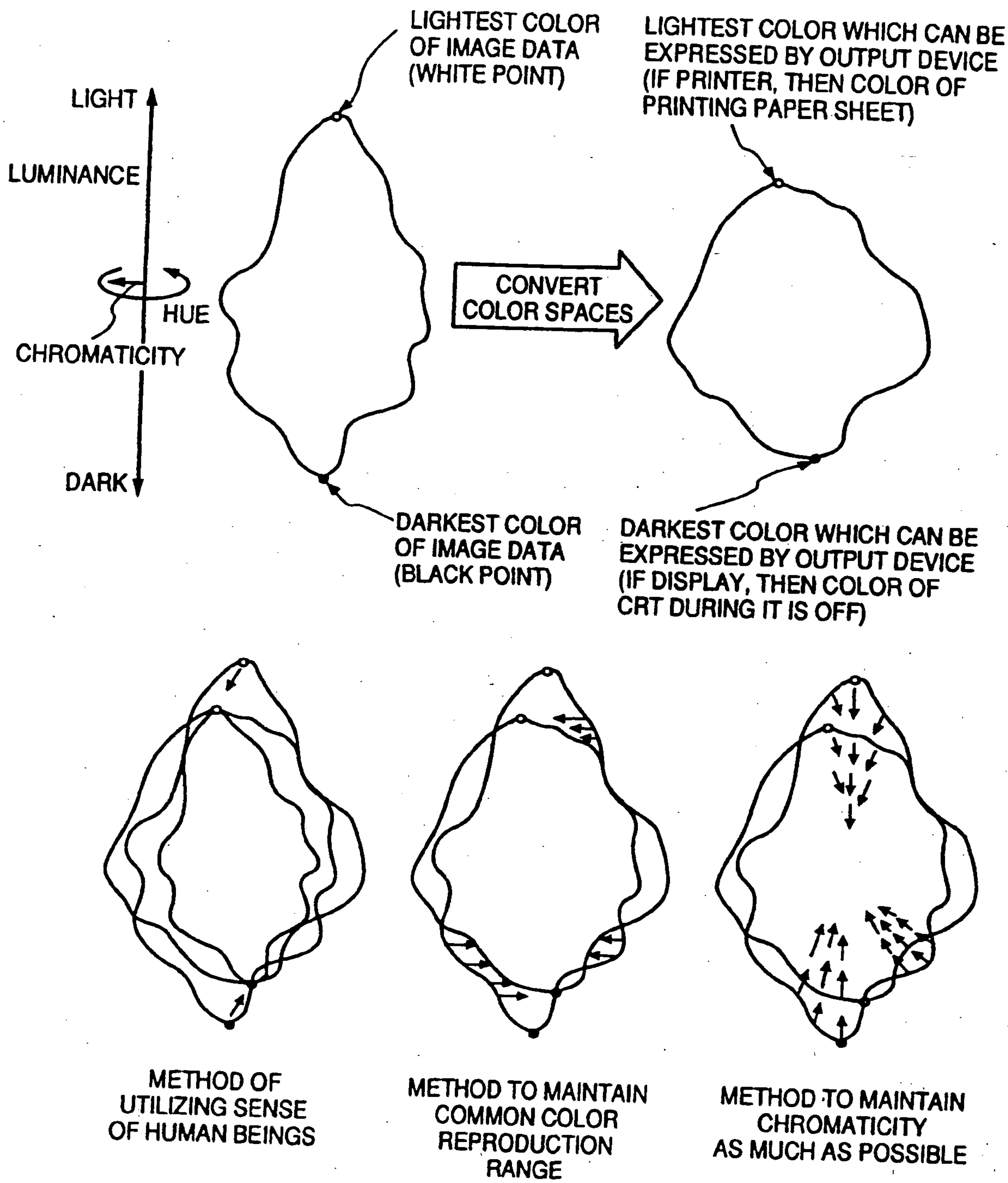

LIGHTEST COLOR
OF IMAGE DATA
(WHITE POINT)
LIGHTEST COLOR WHICH CAN BE
EXPRESSED BY OUTPUT DEVICE
(IF PRINTER, THEN COLOR OF
PRINTING PAPER SHEET)
LIGHT
LUMINANCE
HUE
CHROMATICITY
DARK
CONVERT
COLOR SPACES
DARKEST COLOR
OF IMAGE DATA
(BLACK POINT)
DARKEST COLOR WHICH CAN BE
EXPRESSED BY OUTPUT DEVICE
(IF DISPLAY, THEN COLOR OF
CRT DURING IT IS OFF)
METHOD OF
UTILIZING SENSE
OF HUMAN BEINGS
METHOD TO MAINTAIN
COMMON COLOR
REPRODUCTION
RANGE
METHOD TO MAINTAIN
CHROMATICITY
AS MUCH AS POSSIBLE

FIG. 34